

# Network Security

## Firewalls

# Why Firewalls?

- The Internet allows you access to worldwide resources, but...  
...the Internet also allows the *world* to try and access your resources

# Why Firewalls?

- A **firewall** is inserted between the private network and the Internet
- Provides a **choke point** where security and audits can be imposed
- Single computer system or a set of systems can perform the **firewall function**

# Design Goals

- All traffic, from inside to outside and vice versa, must pass through the firewall
- Only authorized traffic (defined by the security policy) is allowed to flow

# Scope of Firewalls

- **Single choke point** - to protect vulnerable services from various kinds of attack (spoofing, DOS)
- **Platform for non-security functions** – can be used for network address translation and network management
- **Platform for IPSec** – implements VPN via tunnel mode

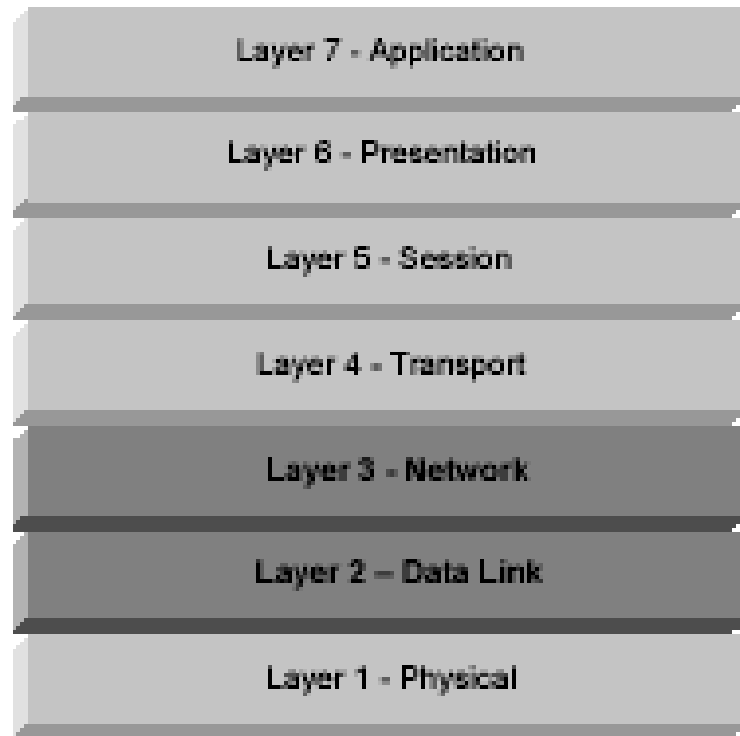
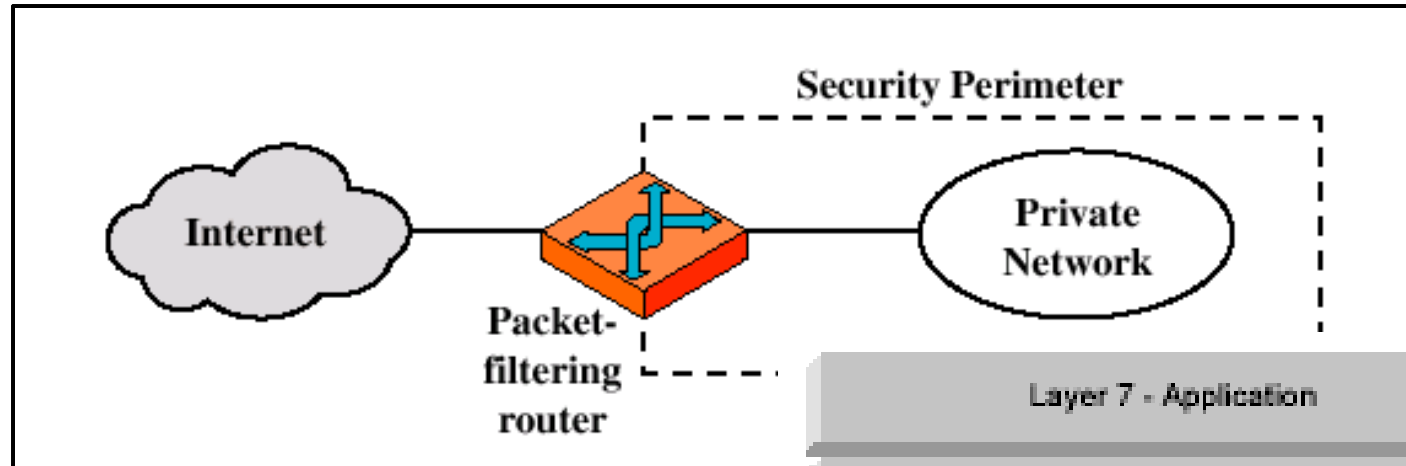
# Limitations of Firewalls

- Cannot protect against attack that bypasses the firewall – **bypass attack**
- Does not protect against **internal threats**
- Cannot protect against the transfer of **virus**-infected programs

# Types of Firewalls

- Packet Filtering Router
- Application Level Gateway
- Circuit Level Gateway

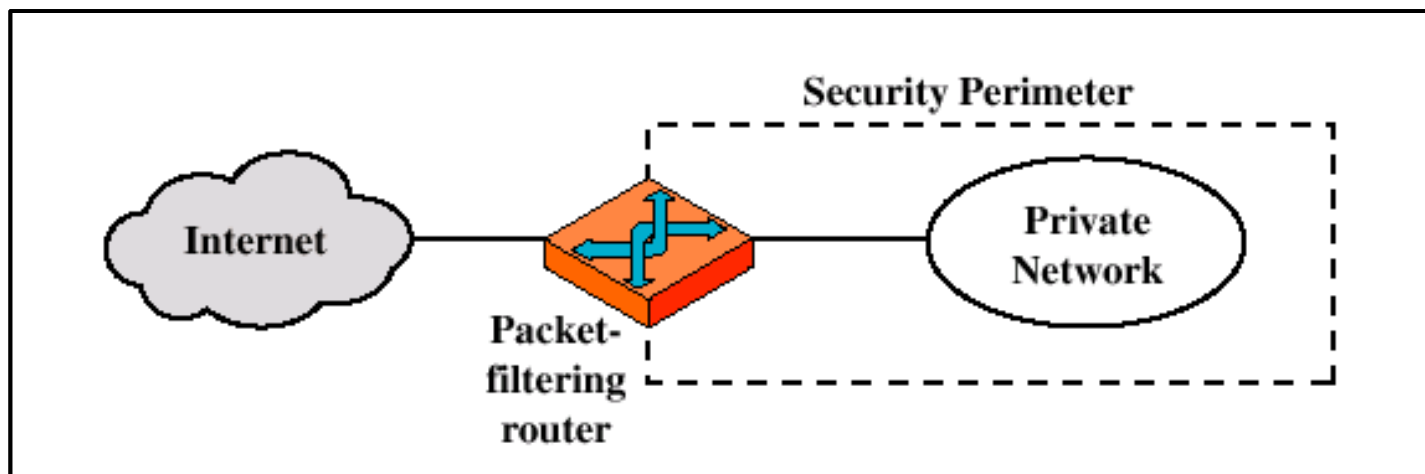
# Packet Filtering





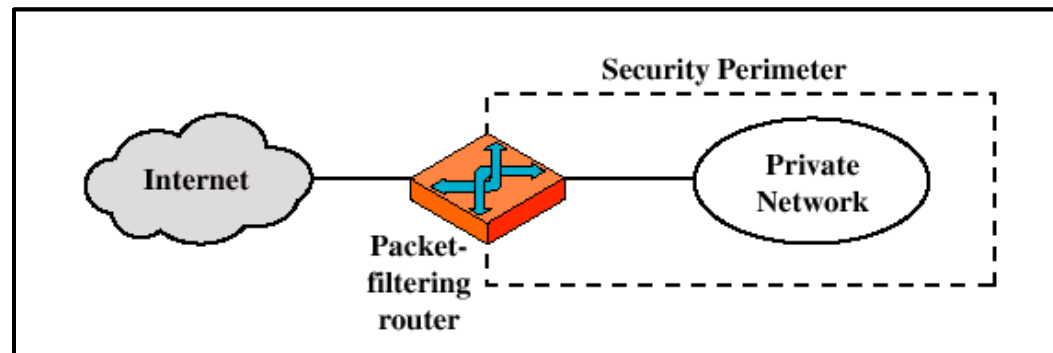
# Packet Filtering Router

- Applies a **set of rules** to each incoming IP packet and *forwards* or *discards* the packet
- Filters packets in *both directions*



# Packet Filtering Router

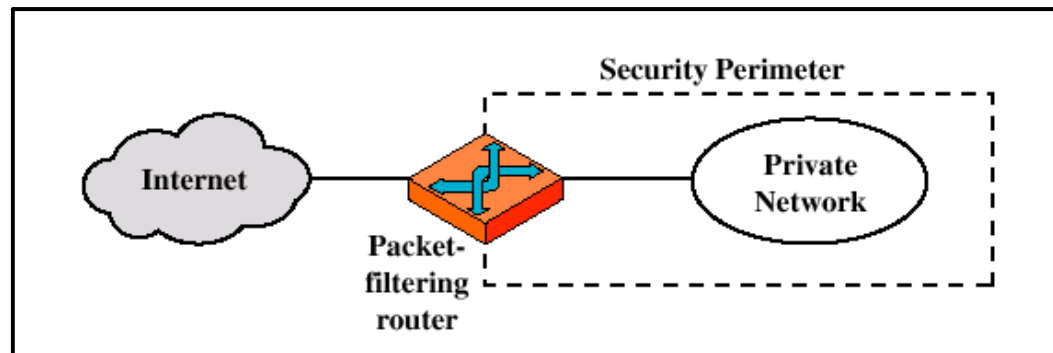
- Rules based on *source* and *destination* address and *port* number and IP protocol and interface
- *List of rules* looking for a match
- If no match, *default* action is taken



# Packet Filtering Router

Two default policies:

- **default = discard:**  
*That which is not expressly permitted is prohibited*
- **default = forward:**  
*That which is not expressly prohibited is permitted*



# Packet Filtering Rules

action	ourhost	port	theirhost	port	comment
block	*	*	SPIGOT	*	we don't trust these guys
allow	OUR-GW	25	*	*	connection to our SMTP port

- Inbound mail is allowed (port 25), but only to a mentioned host
- Everything from SPIGOT is blocked

# Packet Filtering Rules

action	ourhost	port	theirhost	port	comment
block	*	*	*	*	default

- This is the *default policy*
- It is usually the *last* rule
- This rule *drops everything*

# Packet Filtering Rules

action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	Connection to their SMTP port

- Inside host can send mail to the outside
- Some other application could be linked to port 25
- Attacker could gain access through port 25

# Packet Filtering Rules

action	src	port	dest	port	flags	comment
allow	our hosts	*	*	25		connection to their SMTP port
allow	*	25	*	*	ACK	their replies

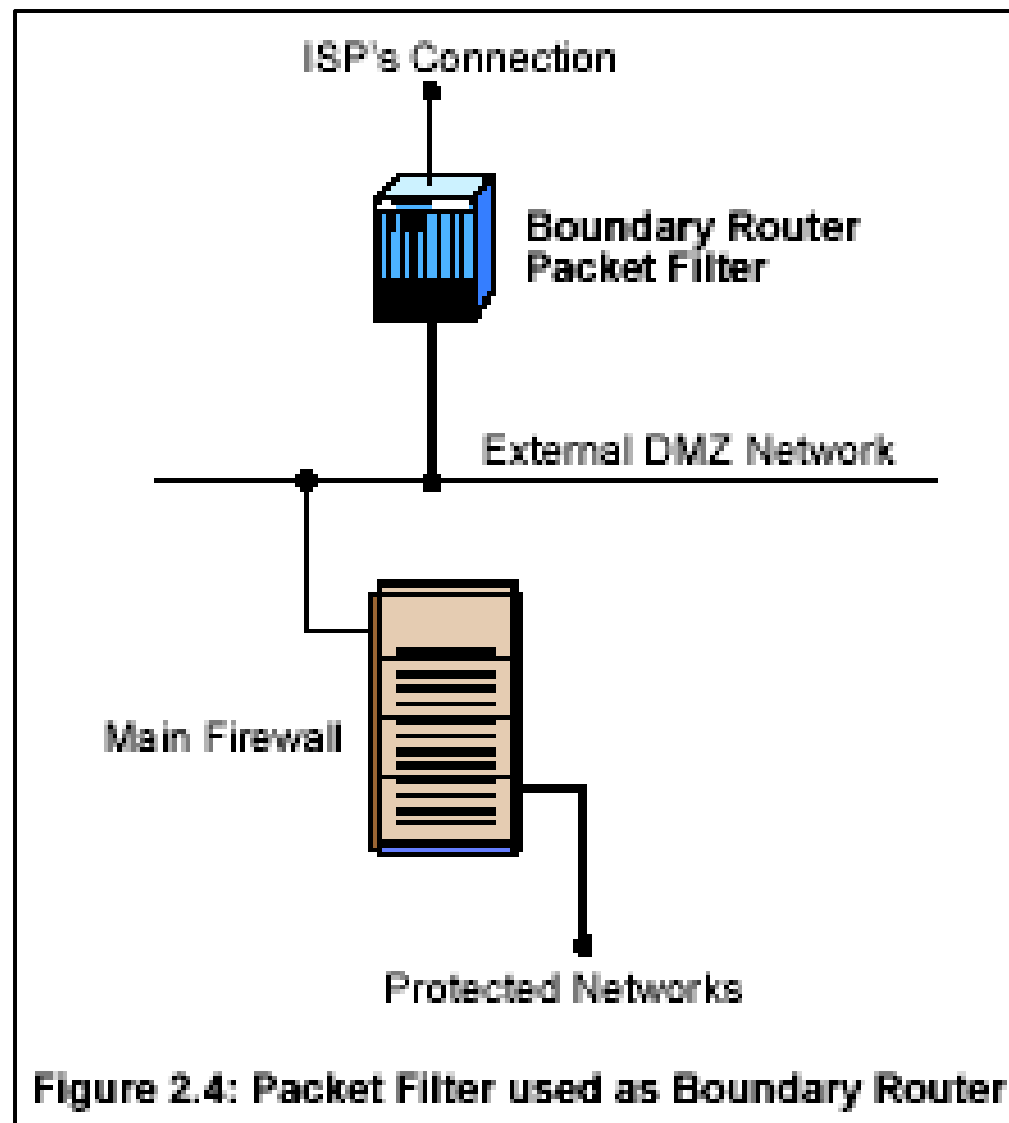
- This improves on the last situation
- Internal hosts can access SMTP anywhere
- ACKs from any SMTP server are permitted

# Packet Filtering

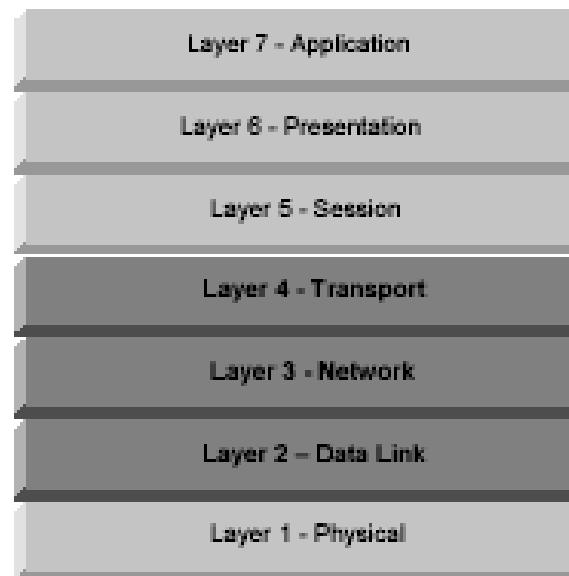
- *Advantage:* simple, transparent and very fast
- *Disadvantage:*
  - Difficulty in setting up rules correctly and completely
  - Vulnerable again Application layer attacks



# Real Life Example



# Stateful Inspection

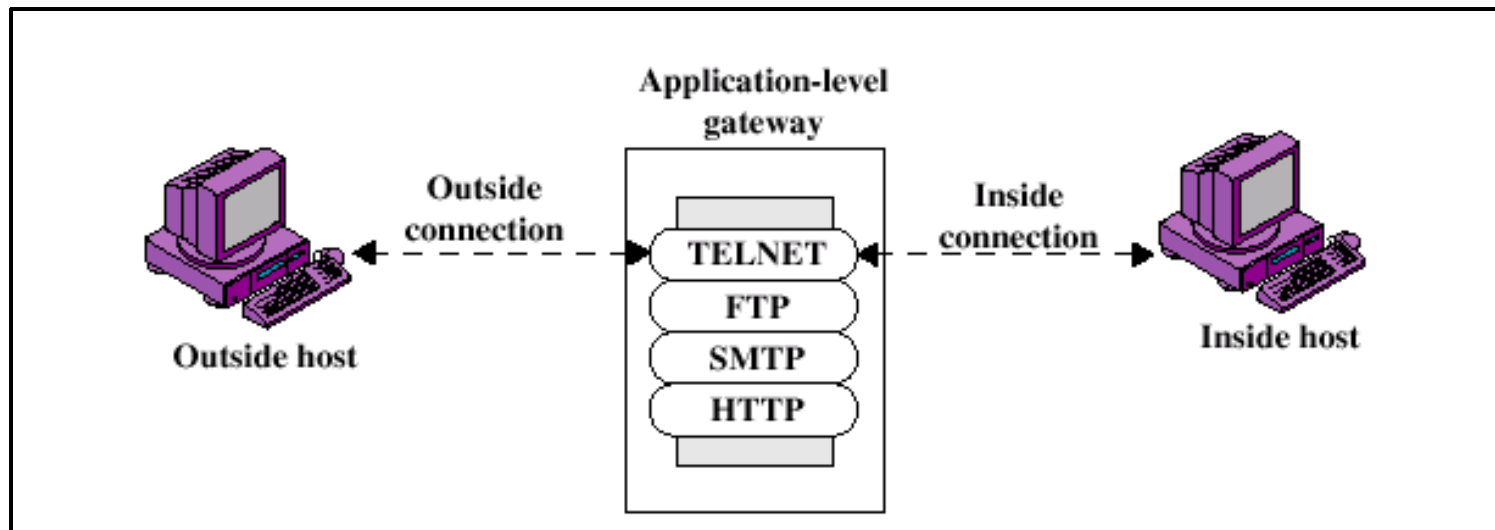


Layers Addressed By Stateful Inspection

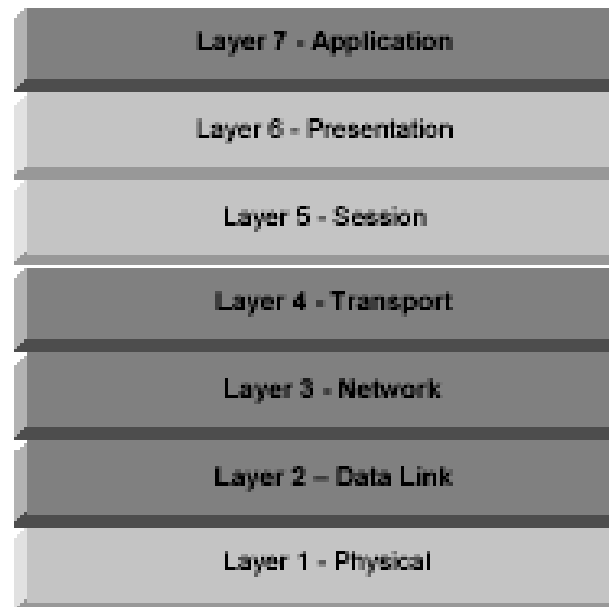
# Stateful Inspection

- **More secure** because the firewall tracks client ports individually rather than opening all high-numbered ports for external access.
- Adds **Layer 4 awareness** to the standard packet filter architecture.

# Application Level Gateway



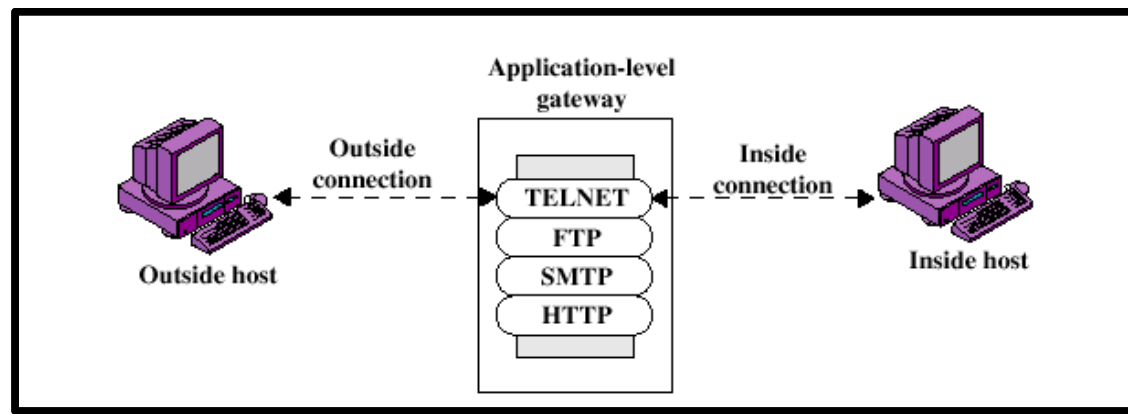
# Application Gateway Firewalls



Layers Addressed by  
Application-Proxy Gateway Firewalls

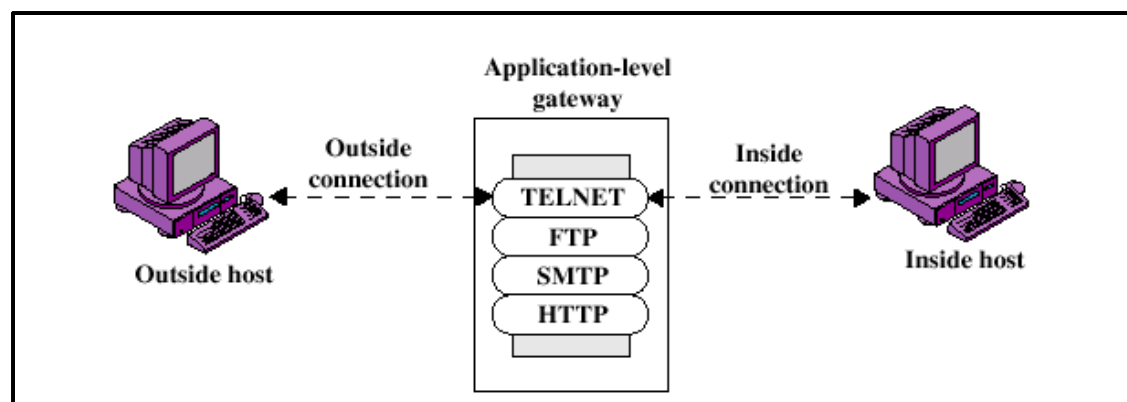
# Application Level Gateway

- Acts as a **relay** of application level traffic
- Also called a **proxy**
- User contacts gateway for TELNET to remote host, user is authenticated, then gateway contacts remote host and relays info between two end points

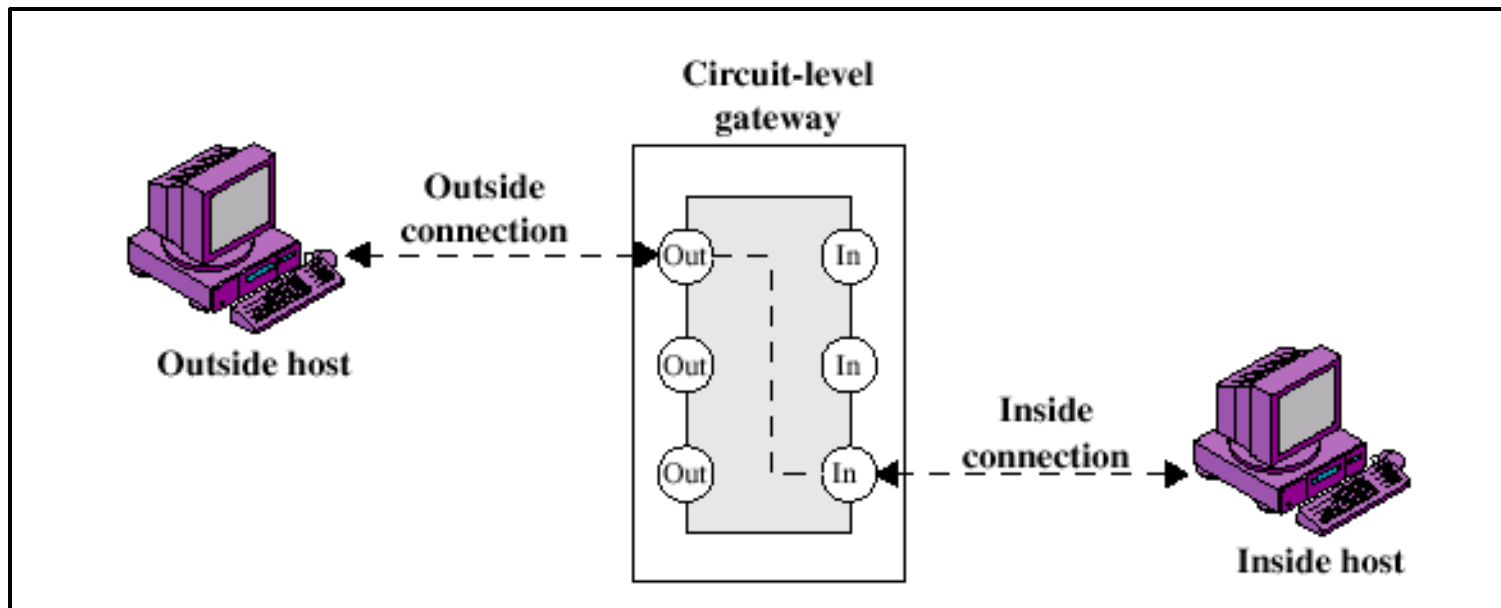


# Application Level Gateway

- Can **examine the packets** to ensure the security of the application – **full packet awareness**
- Very **easy to log** since entire packet seen
- **Disadvantage:** additional processing overhead for each connection – increase load



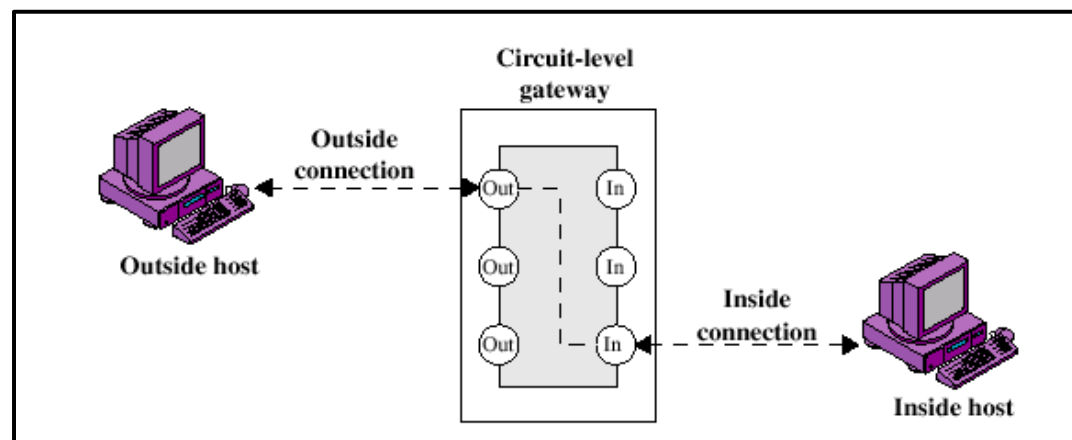
# Circuit-Level Gateway





# Circuit Level Gateway

- *Does not* permit an end-to-end TCP connection
- Sets up *two TCP connections* one between itself and a TCP user on the inside and one between itself and a TCP user on the outside
- *Relays TCP segments* from one connection to the other *without examining the contents*



# Firewall Basing

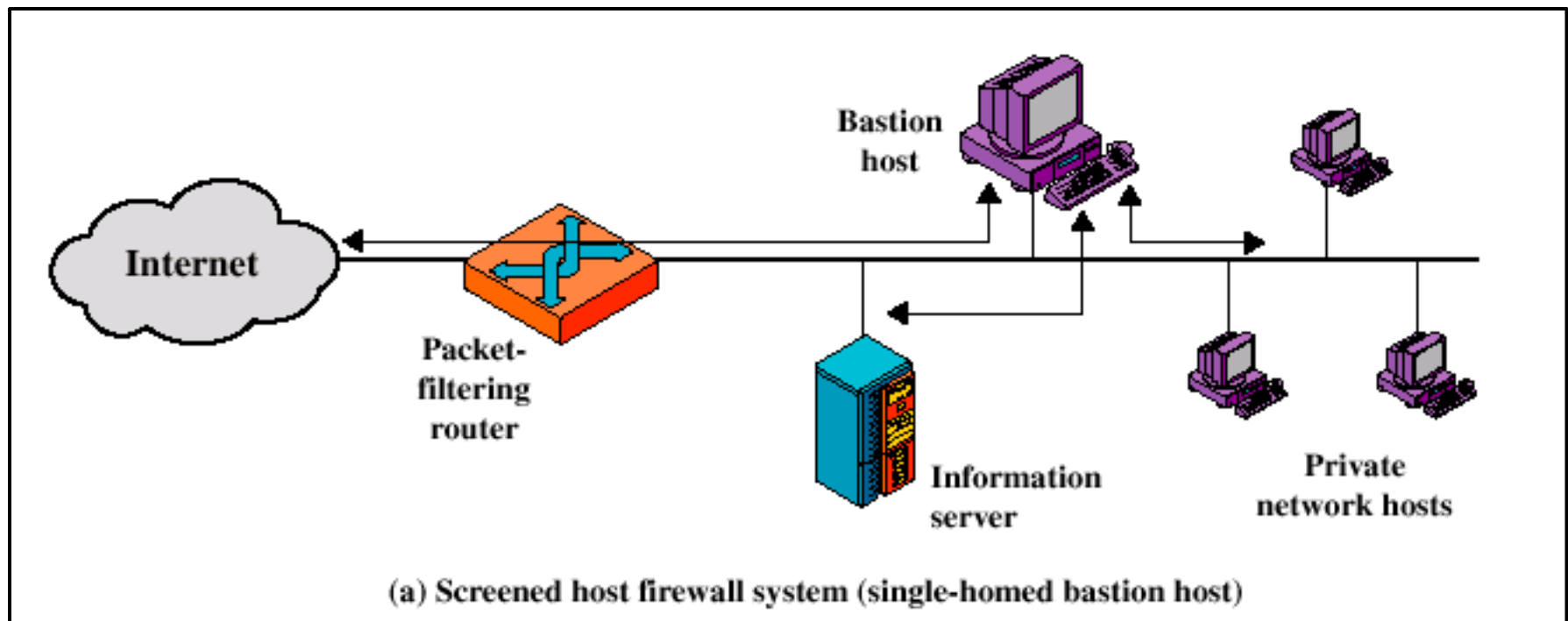
- It is common to base a firewall on a stand-alone machine running a common operating system, such as UNIX or Linux.
- Firewall functionality can also be implemented as a software module in a router or LAN switch

# Bastion Host

- Usually a platform for an application or circuit level gateway
- Only essential services
- Allows *access only* to *specific hosts*
- Maintains detailed *audit information* by logging all traffic
- *Choke point* for discovering and terminating intruder attacks

# Bastion Host, Single-Homed

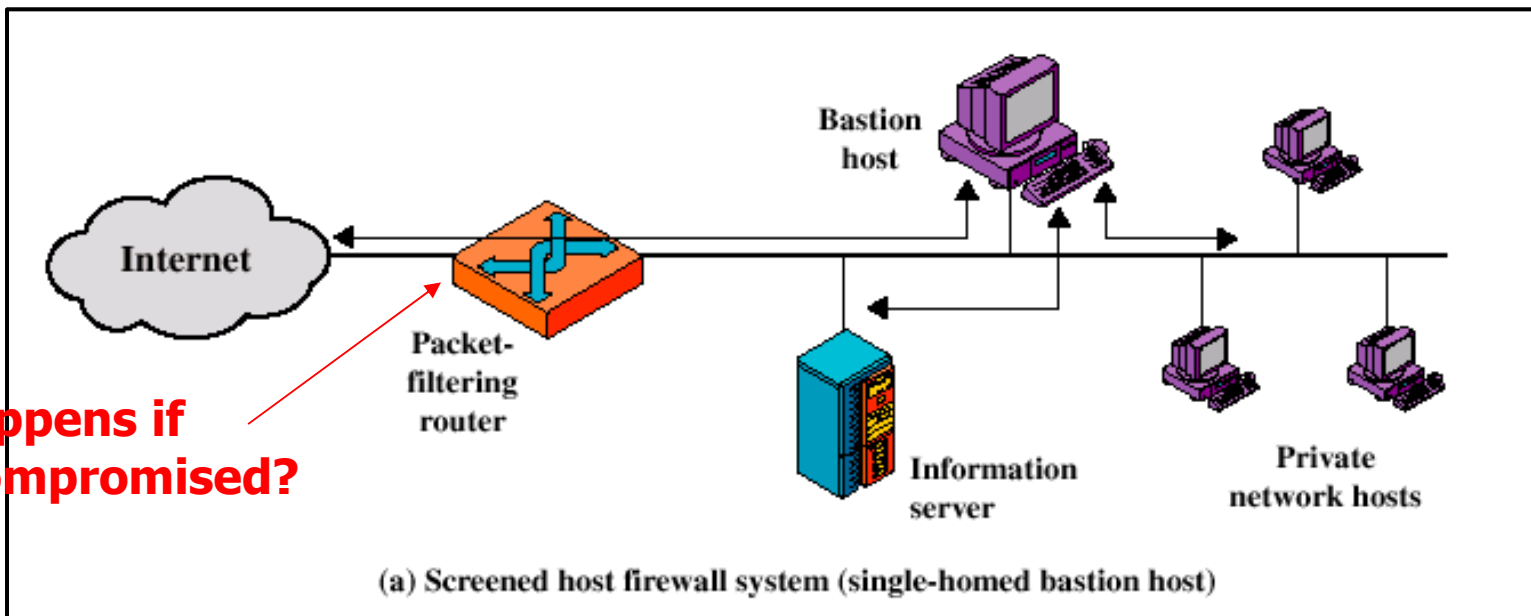
- *Two systems:* packet filtering router and bastion host
- For traffic from the *Internet*, only IP packets *destined* for the *bastion* host are allowed
- For traffic from the *internal network*, only relayed packets *from* the *bastion* host are allowed out



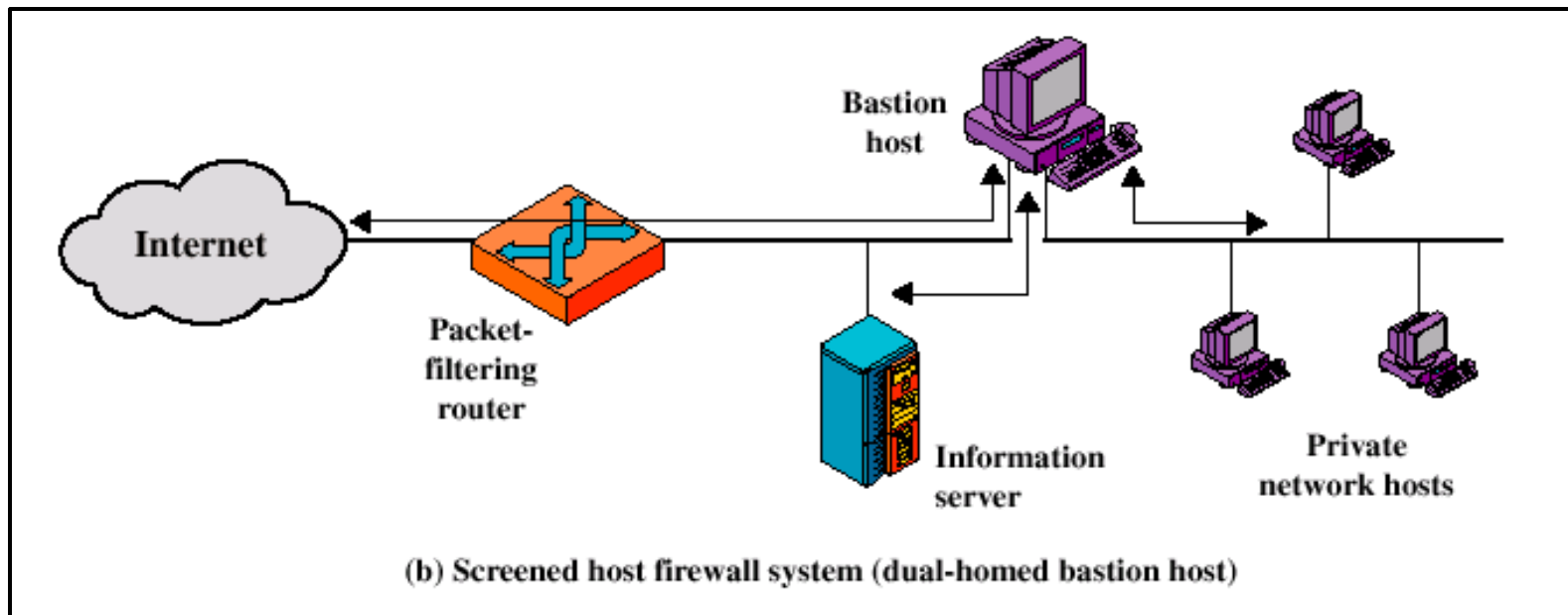
# Bastion Host, Single-Homed

**A Big Problem!!!**

**What happens if  
this is compromised?**

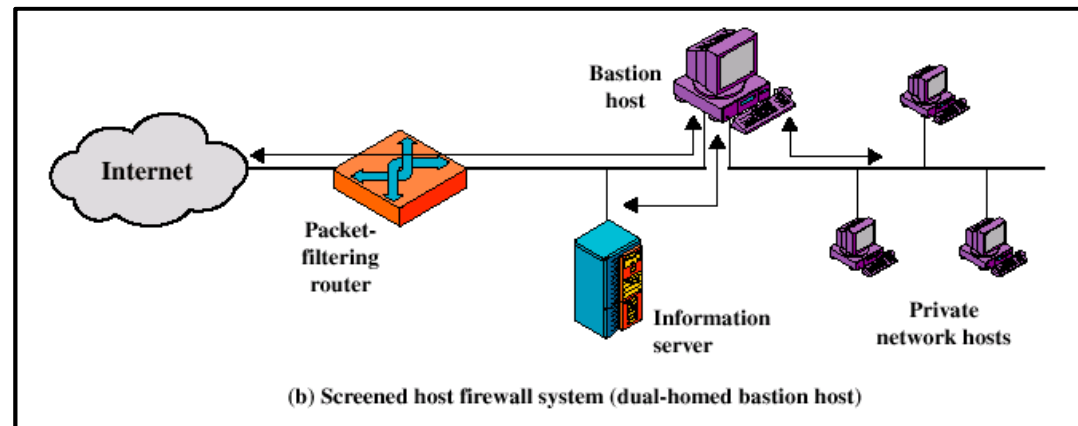


# Bastion Host, Dual-Homed

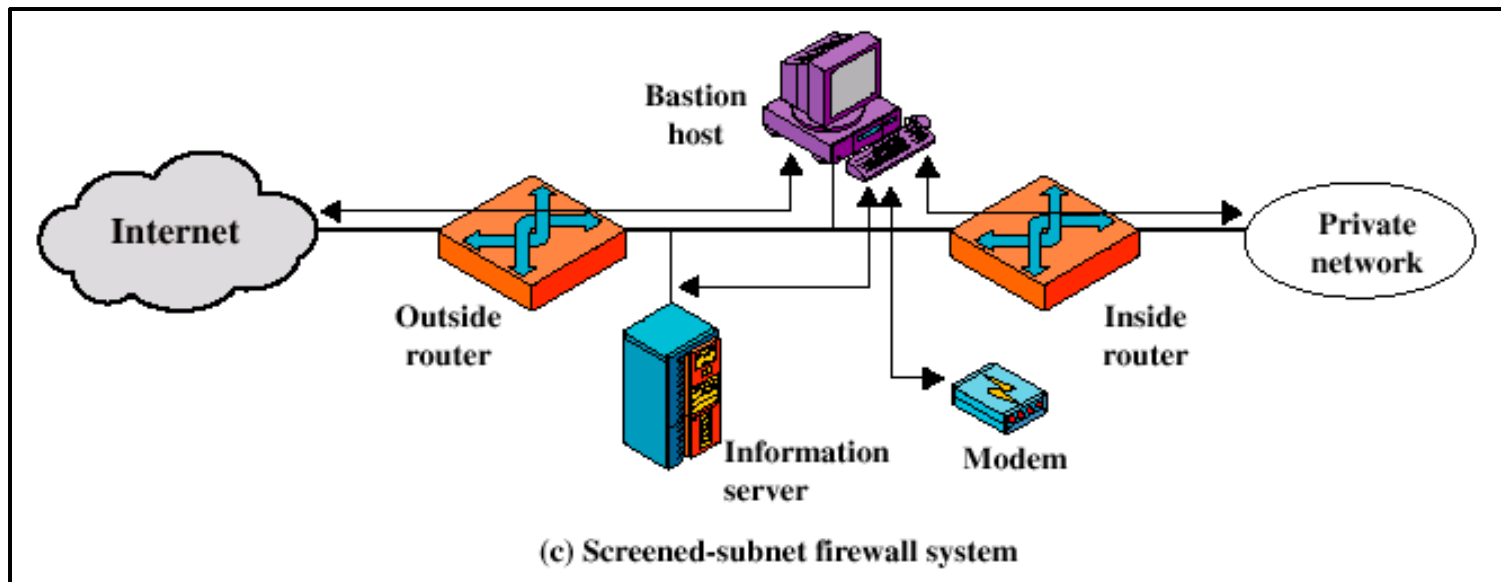


# Bastion Host, Dual-homed

- Bastion host *second defense layer*
- Internal network is completely isolated
- Packet forwarding is turned off
- More secure



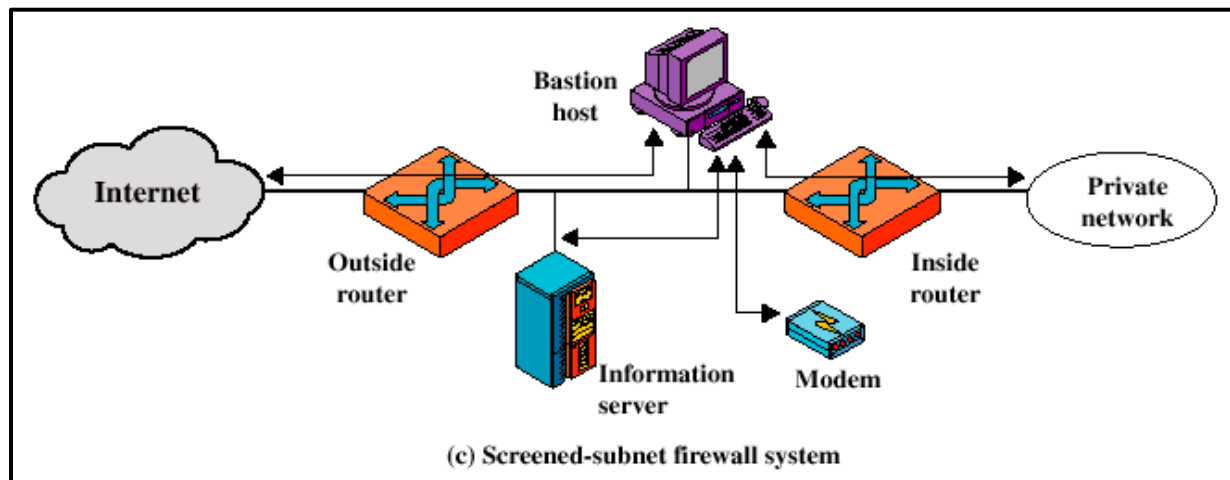
# Screened Subnet



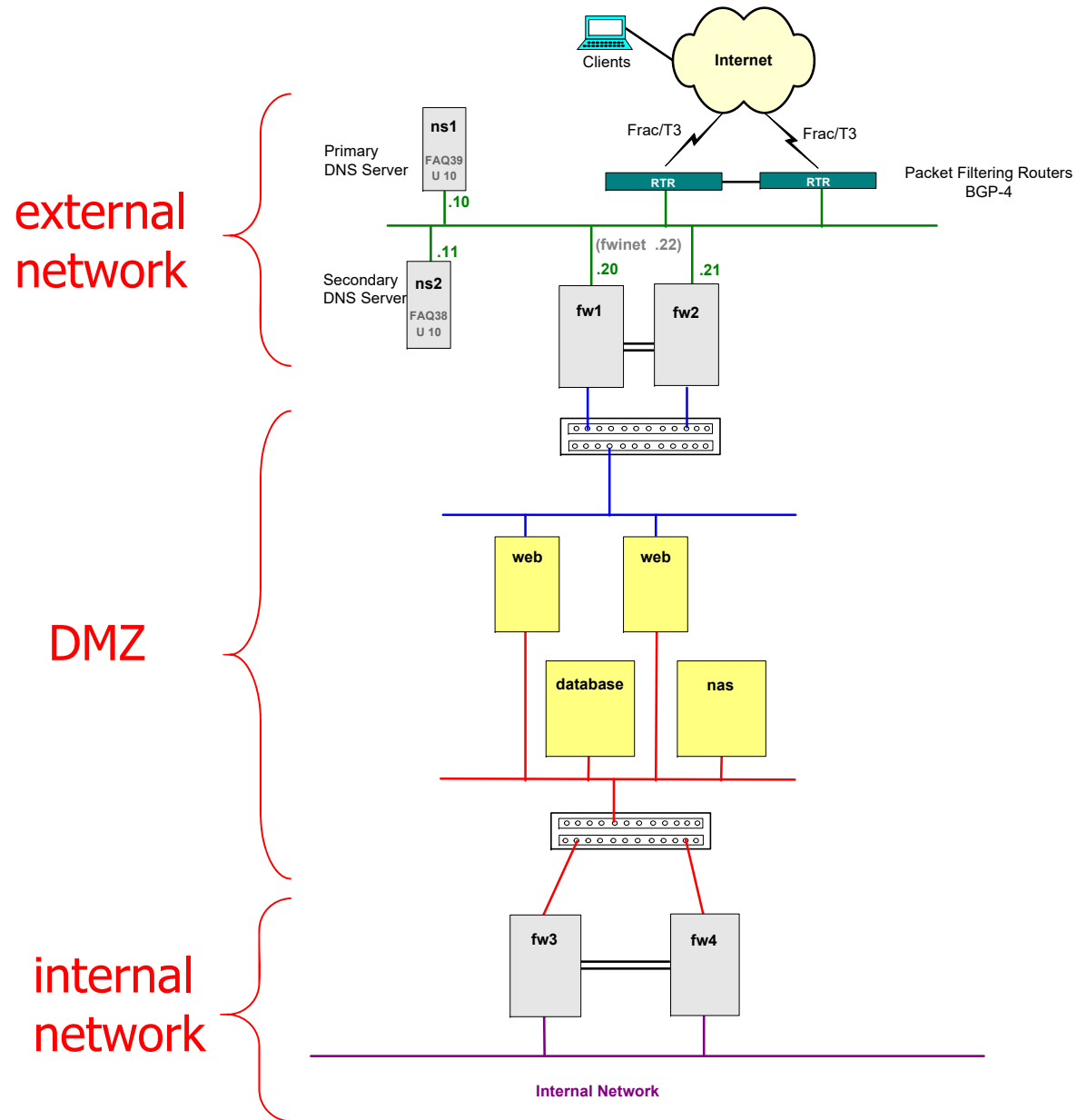


# Screened Subnet

- Most secure
- **Isolated subnet** with bastion host between two packet filtering routers
- **Traffic** across screened subnet is **blocked**
- Three **layers** of defense
- Internal network is **invisible** to the Internet



# Typical DMZ



# What Is iptables?

- Stateful packet inspection.
  - The firewall keeps track of each connection passing through it, This is an important feature in the support of active FTP and VoIP.
- Filtering packets based on MAC address, IPv4 and IPv6
- Filtering packets based the values of the flags in the TCP header
  - Helpful in preventing attacks using malformed packets and in restricting access.
- Network address translation and Port translating NAT/NAPT
  - Building DMZ and more flexible NAT environments to increase security.
- System logging of network activities
  - Provides the option of adjusting the level of detail of the reporting
- A rate limiting feature
  - Helps to block some types of denial of service (DoS) attacks.

# Packet Processing In iptables

- Three builtin **tables** (queues) for processing:
  1. **MANGLE**: is used in QOS to handle marking of packet or in data load distribution to distribute packets to different routes
  2. **FILTER**: packet filtering, has three builtin chains (your firewall policy rules)
    - Forward chain**: filters packets to servers protected by firewall
    - Input chain**: filters packets destined for the firewall
    - Output chain**: filters packets originating from the firewall
  3. **NAT**: network address translation, has two builtin chains
    - Pre-routing**: NAT packets when destination address need changes
    - Post-routing**: NAT packets when source address need changes

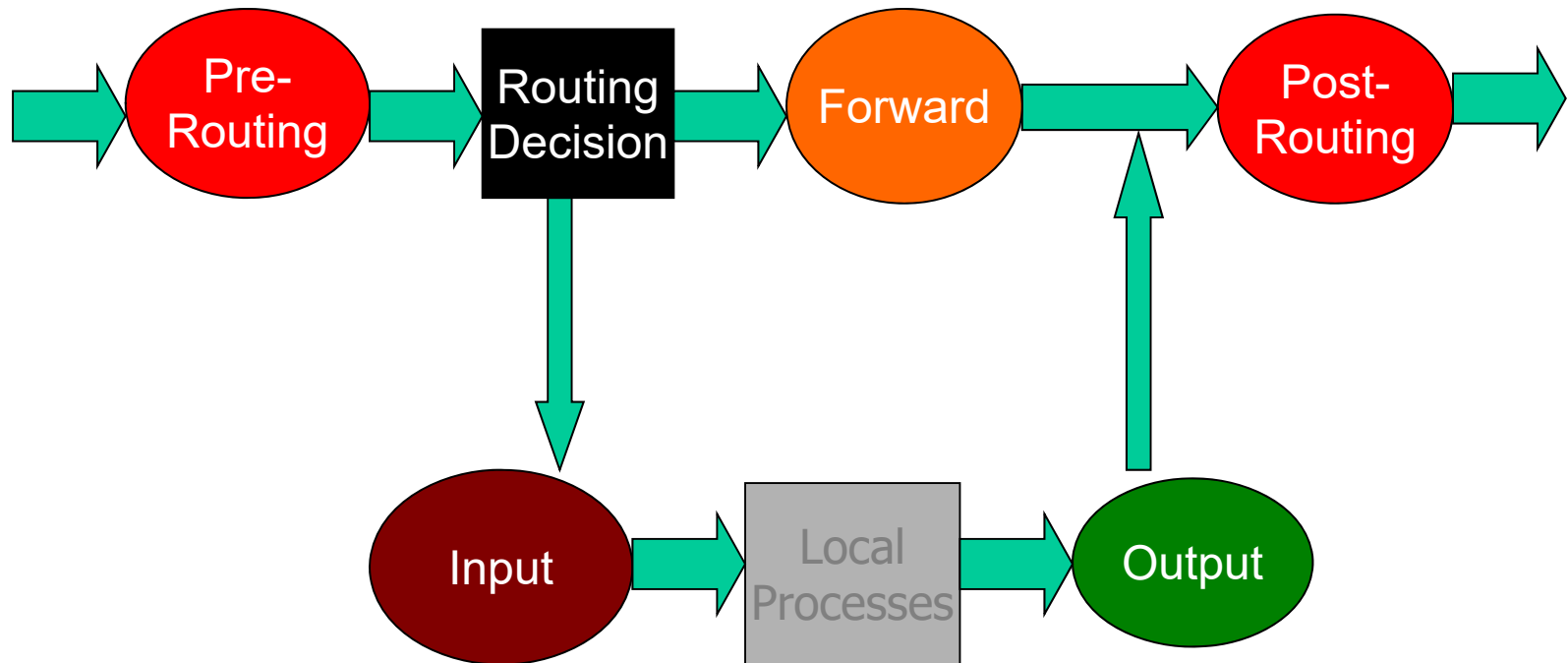
# Processing For Packets Routed By The Firewall 1/2

Queue Type	Queue Function	Packet transformation chain in Queue	Chain Function
Filter	Packet filtering	FORWARD	Filters packets to servers accessible by another NIC on the firewall.
		INPUT	Filters packets destined to the firewall.
		OUTPUT	Filters packets originating from the firewall
Nat	Network Address Translation	PREROUTING	Address translation occurs before routing. Facilitates the transformation of the destination IP address to be compatible with the firewall's routing table. Used with NAT of the destination IP address, also known as <b>destination NAT</b> or <b>DNAT</b> .

## Processing For Packets Routed By The Firewall 2/2

		POSTROUTING	Address translation occurs after routing. This implies that there was no need to modify the destination IP address of the packet as in pre-routing. Used with NAT of the source IP address using either one-to-one or many-to-one NAT. This is known as <b>source NAT</b> , or <b>SNAT</b> .
		OUTPUT	Network address translation for packets generated by the firewall. (Rarely used in SOHO environments)
Mangle	TCP header modification	PREROUTING POSTROUTING OUTPUT INPUT FORWARD	Modification of the TCP packet quality of service bits before routing occurs (Rarely used in SOHO environments)

# Packet Traversal in Linux



# IPtables “chains”

- A *chain* is a sequence of filtering rules.
- Rules are checked in order. First match wins. Every chain has a default rule.
- If no rules match the packet, default policy is applied.
- Chains are dynamically inserted/deleted.