Advanced Network Security

Firewalls and IDS

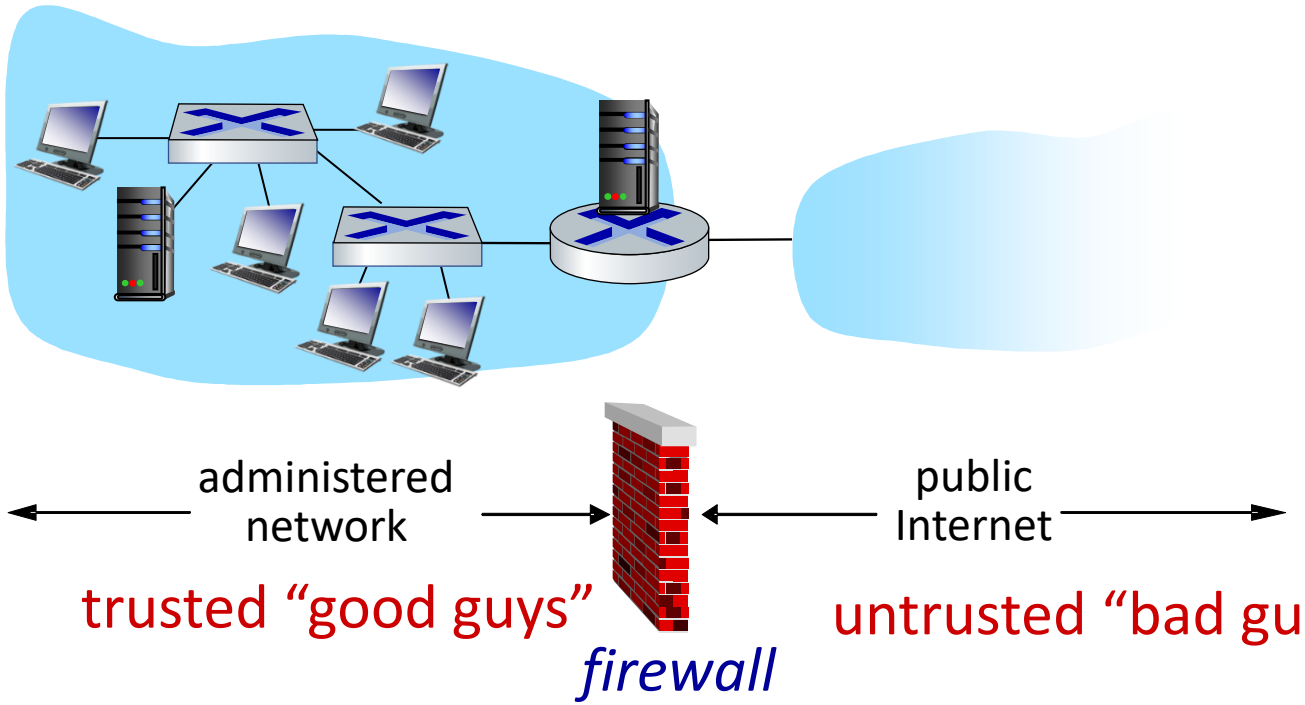
Amir Mahdi Sadeghzadeh, Ph.D.

Most slides have been adapted from Steven Bellovin, COMS W4180, Columbia University, 2006, and Mehdi Kharrazi, CE40-817, Sharif University of Technology, 2015

Firewalls

firewall

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



What's a Firewall

- Barrier between us and them.
- Limits communication to the outside world.
- The outside world can be another part of the same organization.
  - Only a very few machines exposed to attack.

- Major firewall vendors: checkpoint, Cisco PIX/ASA

Why Use Firewalls?

- Most **hosts have security holes**.
  - Most **software is buggy**. Therefore, most security software has security bugs.
- Firewalls run much **less code**, and hence have **few bugs** (and holes).
- Firewalls can be professionally (and hence better) **administered**.
- Firewalls run less software, with **more logging and monitoring**.
- They enforce the partition of a network into **separate security domains**.
  - Without such a partition, a network acts as a **giant virtual machine**, with an unknown set of privileged and ordinary users.

Traditional Firewalls by Analogy

- Passports are (generally) checked at the border.
- My office doesn’t have a door direct to the outside.
- Any way, what does “firewall” mean? Where does the name come from?

Should We Fix the Network Protocols Instead?

- Network security is not the problem.
- Firewalls are not a solution to network problems. They are a network response to a host security problem.
- More precisely, they are a response to the dismal state of software engineering; taken as a whole, the profession does not know how to produce software that is secure, correct, and easy to administer.
- Consequently, better network protocols will not obviate the need for firewalls.

The best cryptography in the world will not guard against buggy code.

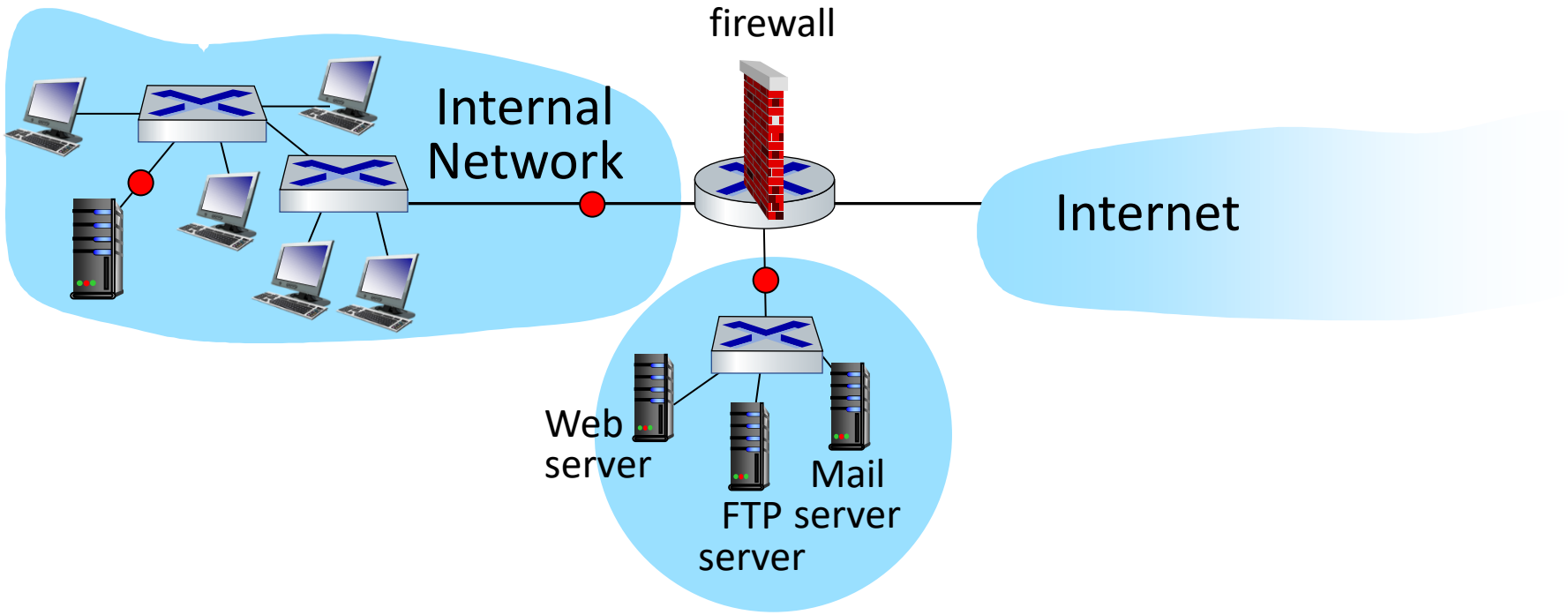
Firewall Advantages

- If you don’t need it, get rid of it.
  - No ordinary users, and hence no passwords for them
  - Run as few servers as possible
  - Install conservative software, don’t get the latest fancy servers, etc.
  - Log everything, and monitor the log files.
  - Keep copious backups, including a “Day 0” backup.
- Ordinary machines cannot be run that way.

What a firewall cannot do

- Cannot protect you against malicious insiders.
- Cannot protect against connections that do not pass through it.
- Cannot protect against completely new threats.

Schematic of a Firewall



DMZ (Demilitarized Zone)

Conceptual Pieces

- An “inside” — everyone on the inside is presumed to be a good guy
- An “outside” — bad guys live there
- A “DMZ” (Demilitarized Zone) — put necessary but potentially dangerous servers there

The DMZ

- Good spot for things like mail and web servers
- Outsiders can send email, retrieve web pages
- Insiders can retrieve email, update web pages
- Must monitor such machines very carefully!

Firewall Philosophies

- Black list
  - Block all dangerous destinations.
- White list
  - Block everything; unblock things known to be both safe and necessary.

- Option 1 gets you into an arms race with the attackers; you have to know everything that is dangerous, in all parts of your network. Option 2 is much safer.

Blocking Outbound Traffic?

- Many sites permit arbitrary outbound traffic, but. . .
  - Internal bad guys?
  - Extrusion detection?
  - Regulatory requirements?
  - Other corporate policy?

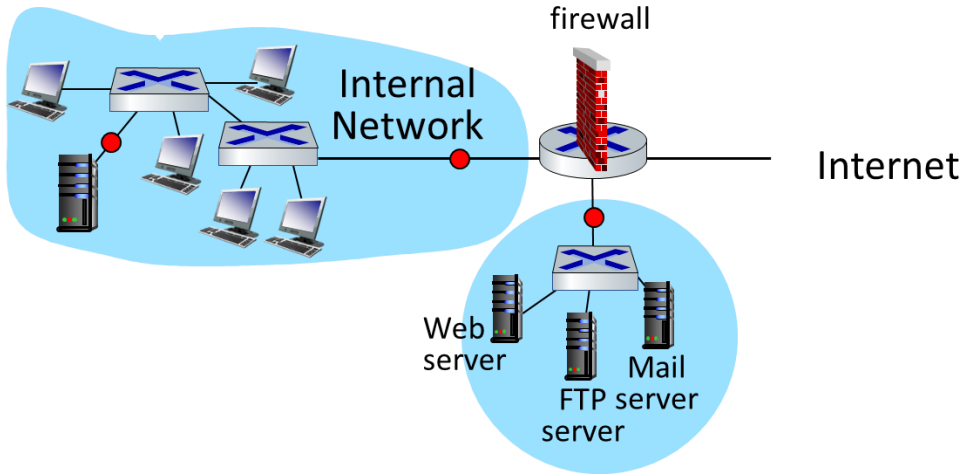
Types of Firewalls

- (Stateless) Packet Filters
- Stateful Packet Filters
- Application Gateways
  - Web Application Firewalls (WAFs)
- Circuit Relays
- Personal and/or Distributed Firewalls

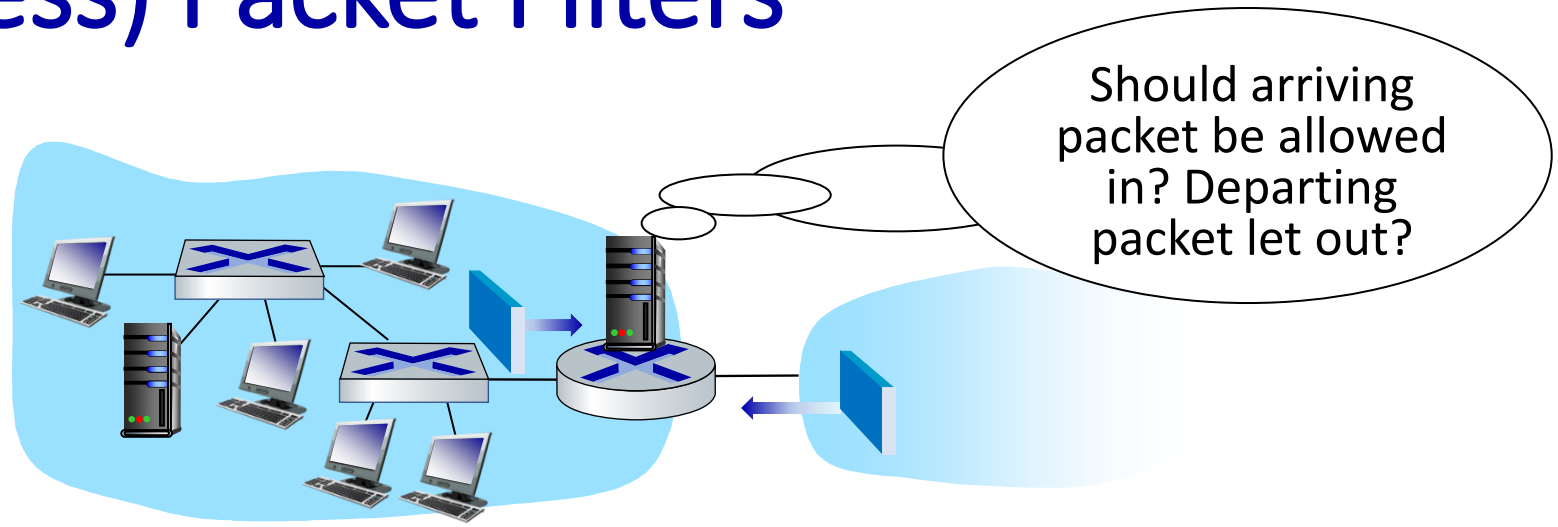
- Many firewalls are combinations of these types.

Types of Firewalls

- (Stateless) Packet Filters
- Stateful Packet Filters
- Application Gateways
- Circuit Relays
- Personal and/or Distributed Firewalls



(Stateless) Packet Filters



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

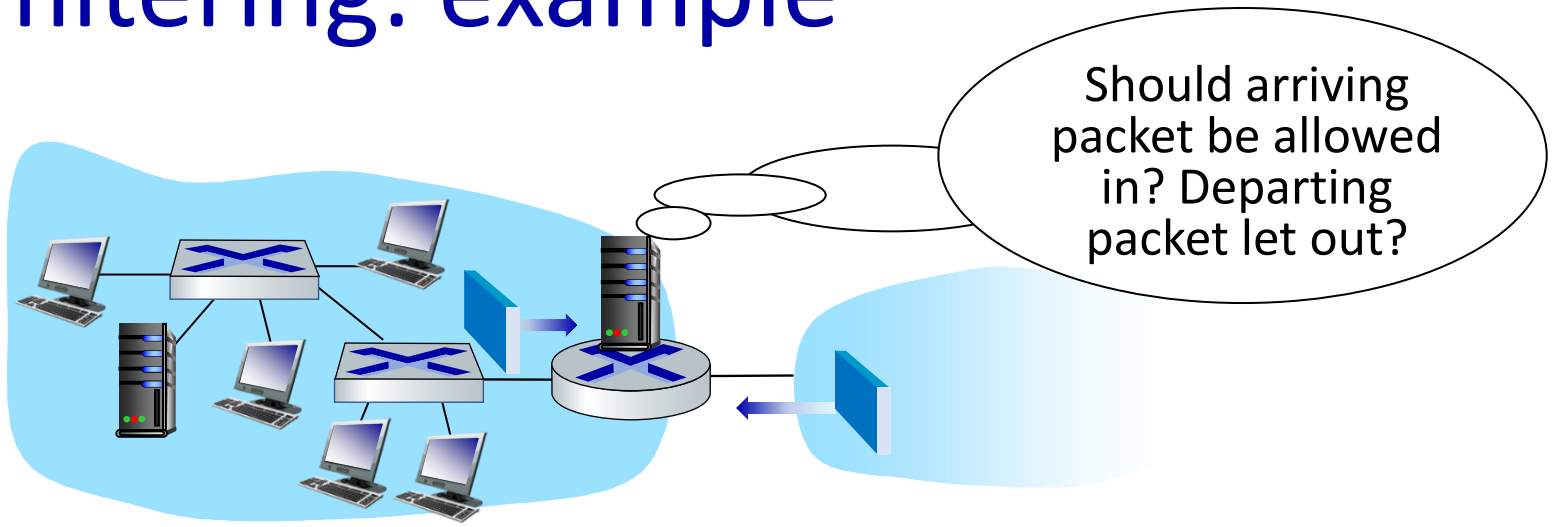
Packet Filters

- Router-based (and hence cheap).
- Individual packets are accepted or rejected; no context is used.
- Filter rules are hard to set up; the primitives are often inadequate, and different rules can interact.
- Packet filters a poor fit for ftp similar services.

Running Without State

- We want to permit outbound connections
- We have to permit reply packets
- For TCP, this can be done without state
  - The very first packet of a TCP connection has just the SYN bit set
  - All others have the ACK bit set
  - Solution: allow in all packets with ACK turned on

Packet filtering: example



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

Stateless packet filtering: more examples

Policy	Firewall Setting
no outside Web access	drop all outgoing packets to any IP address, port 80
no incoming TCP connections, except those for institution's public Web server only.	drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
prevent Web-radios from eating up the available bandwidth.	drop all incoming UDP packets - except DNS and router broadcasts.
prevent your network from being used for a smurf DoS attack.	drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255)
prevent your network from being tracerouted	drop all outgoing ICMP TTL expired traffic

Firewall Rules Setup

- Action:
  - Permit (Pass) Allow the packet to proceed
  - Deny (Block) Discard the packet
- Direction:
  - Source (where the packet comes from) <IP Address, Port> or network
  - Destination (where the packet goes) <IP Address, Port> or network
- Protocol:
  - TCP, UDP
- Packet Flags:
  - ACK, SYN, RST, etc.

Sample Rule Set

- We want to block a spammers, but allow anyone else to send email to our gateway.

block: theirhost = spammer

allow: theirhost = any **and**

theirport = any **and**

ourhost = our-gw **and**

ourport = 25.

Sample Rule Set

- We want to allow all conversations with remote mail gateways.

Allow: theirhost = any **and**
theirport = 25 **and**
ourhost = any **and**
ourport = any.

- Problem?
  - We don’t control port number selection on the remote host. Any remote process on port 25 can call in.

The Right Choice

- Permit outgoing calls.

Allow: theirhost = any **and**
theirport = 25 **and**
ourhost = any **and**
ourport = any **and**
bitset(ACK).

Packet Filters and UDP

- UDP has no notion of a connection. It is therefore impossible to distinguish a reply to a query—which should be permitted—from an intrusive packet.
- Address-spoofing is easy — no connections
- At best, one can try to block known-dangerous ports. But that’s a risky game.
- The safe solution is to permit UDP packets through to known-safe servers only.

UDP Example: DNS

- Accepts queries on port 53
- Block if handling internal queries only; allow if permitting external queries

ICMP Problems

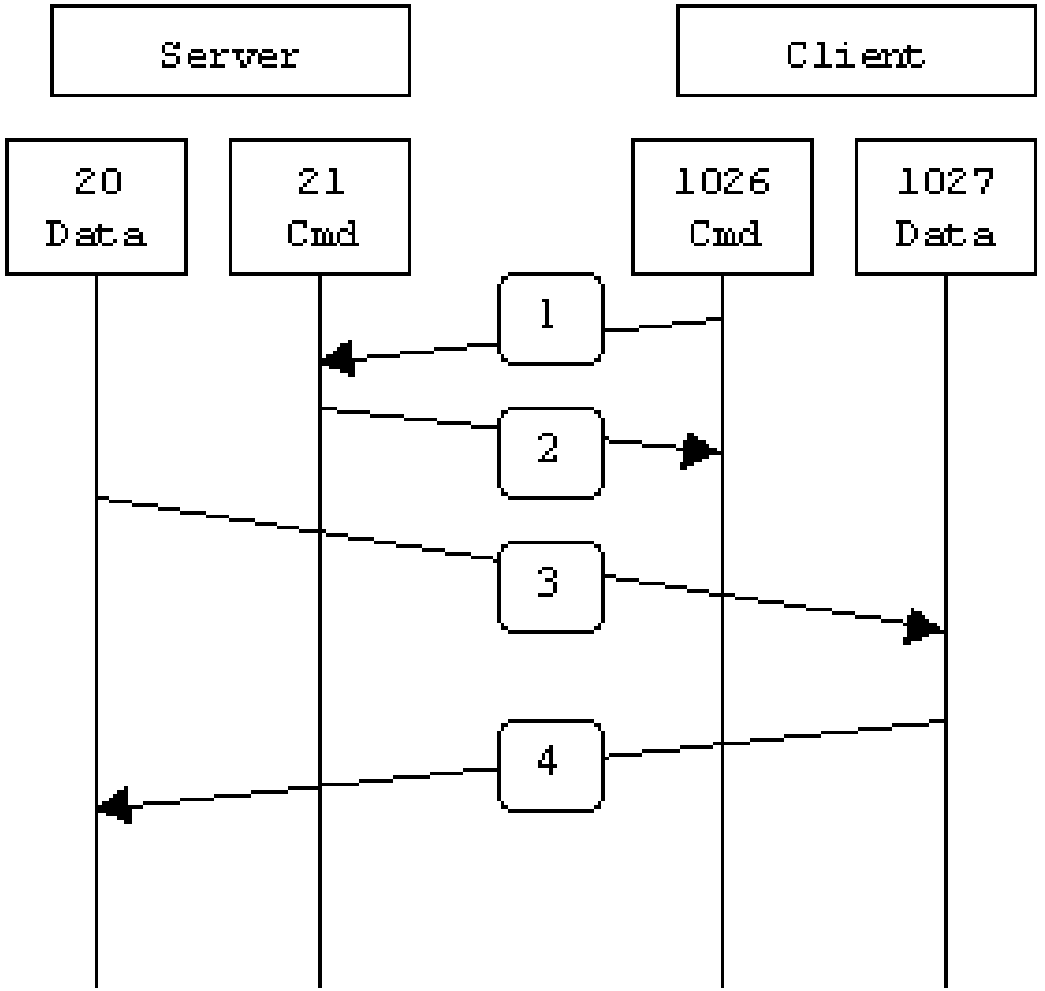
- Often see ICMP packets in response to TCP or UDP packets
- Important example: “Path MTU” response
  - Must be allowed in or connectivity can break
- Simple packet filters can’t match things up

FTP

- FTP clients (and some other services) use secondary channels
- Again, these live on random port numbers
- Simple packet filters cannot handle this

FTP

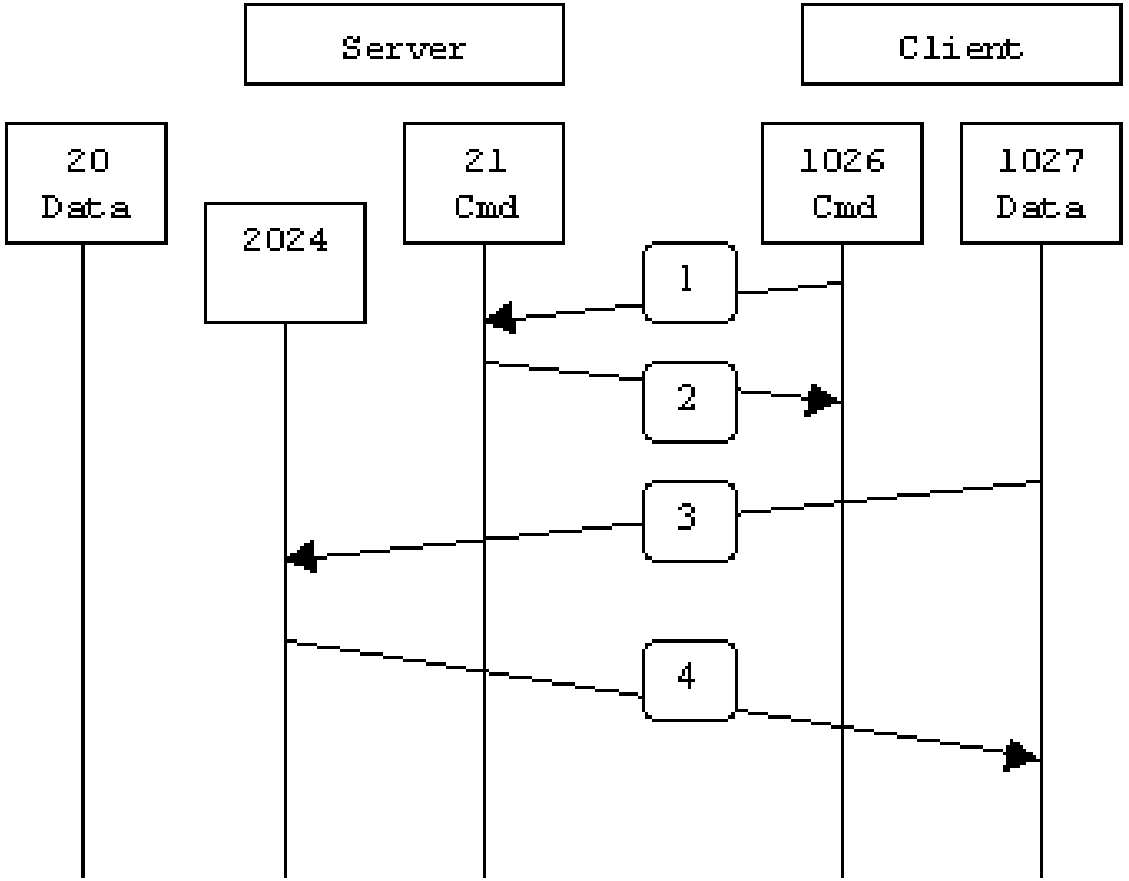
- In active mode FTP:
  - Client connects from a random unprivileged port (N > 1023) to the FTP server's command port, port 21.
  - Client starts listening to port N+1 and sends the FTP command PORT N+1 to the FTP server.
  - Server connects back to the client's specified data port from its local data port, which is port 20.
- Problem?



[http://slacksite.com/other/ftp.html]

FTP

- Passive mode FTP:
  - Client opens two random unprivileged ports locally (N > 1023 and N+1).
  - The first port contacts the server on port 21
  - Instead of then issuing a PORT command, client issues the PASV command.
  - Server opens a random unprivileged port (P > 1023) and sends the PORT P command back to the client.
  - Client then initiates the connection from port N+1 to port P on the server to transfer data.



[http://slacksite.com/other/ftp.html]

Saving FTP

- By default, FTP clients send a PORT command to specify the address for an inbound connection
- If the PASV command is used instead, the data channel uses a separate outbound connection
- If local policy permits arbitrary outbound connections, this works well

The Role of Packet Filters

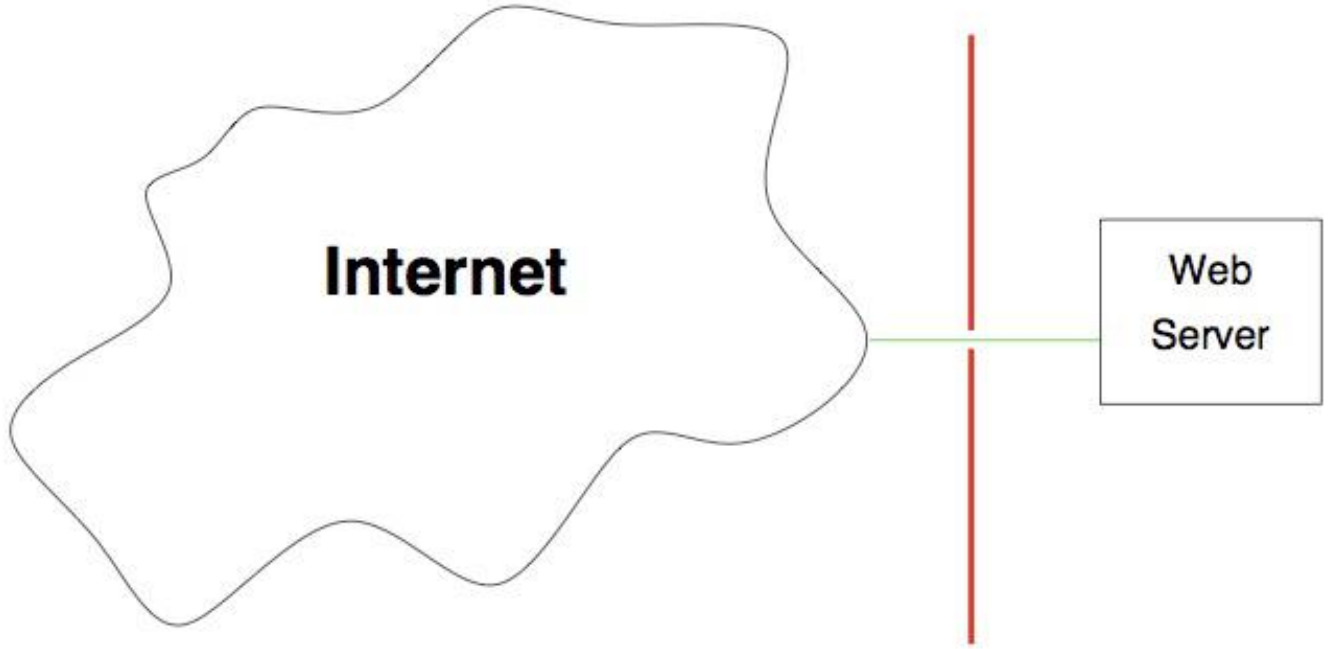
- Packet filters are not very useful as general-purpose firewalls
- That said, they have their place
  - Several special situations where they're perfect

Simplicity

- Packet filters are very simple, and can protect some simple environments
- Virtually all routers have the facility built in

Point Firewalls

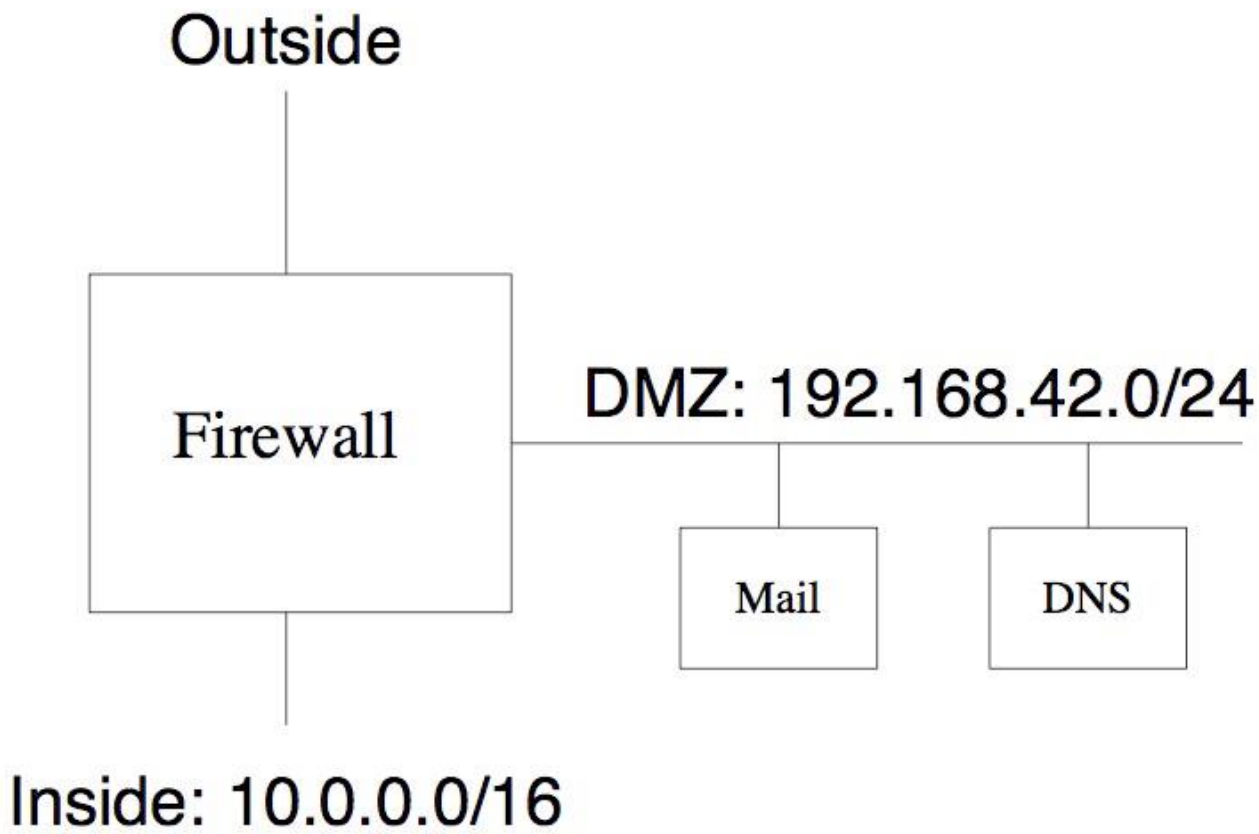
- Allow in ports 80 and 443. Block everything else. This is a Web server appliance — it shouldn't do anything else! But — it may have necessary internal services for site administration.



Address Filtering

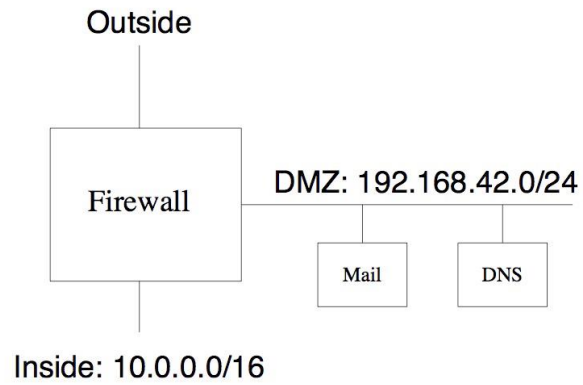
- At the border, block internal addresses from coming in from the outside
- Similarly, prevent fake addresses from going out

Sample Configuration



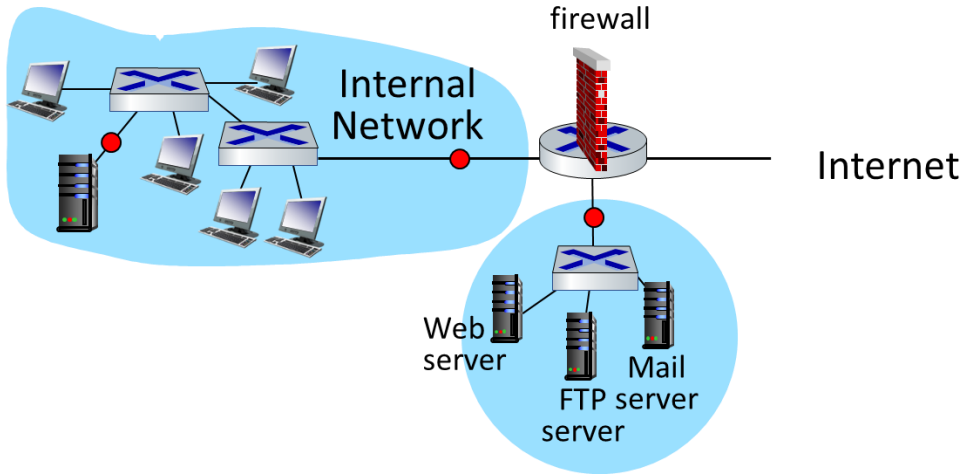
Sample Rules

Interface	Action	Addr	Port	Flags
Outside	Block	src=10.0.0.0/16		
Outside	Block	src=192.168.42.0/24		
Outside	Allow	dst=Mail	25	
Outside	Block	dst=DNS	53	
Outside	Allow	dst=DNS	UDP	
Outside	Allow	Any		ACK
Outside	Block	Any		
DMZ	Block	src !=192.168.42.0/24		
DMZ	Allow	dst=10.0.0.0/16		ACK
DMZ	Block	dst=10.0.0.0/16		
DMZ	Allow	Any		
Inside	Block	src !=10.0.0.0/16		
Inside	Allow	dst=Mail	993	
Inside	Allow	dst=DNS	53	
Inside	Block	dst=192.168.42.0/24		
Inside	Allow	Any		



Types of Firewalls

- (Stateless) Packet Filters
- Stateful Packet Filters
- Application Gateways
- Circuit Relays
- Personal and/or Distributed Firewalls



Stateful packet filtering

- *stateless packet filter:* heavy handed tool
  - admits packets that "make no sense," e.g., dest port = 80, ACK bit set, even though no TCP connection established:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- *stateful packet filter:* track status of every TCP connection
  - track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets "**makes sense**"
  - timeout inactive connections at firewall: no longer admit packets

Stateful Packet Filters

- Most common type of packet filter
- Solves many — but not all — of the problems with simple packet filters
- Requires per-connection state in the firewall

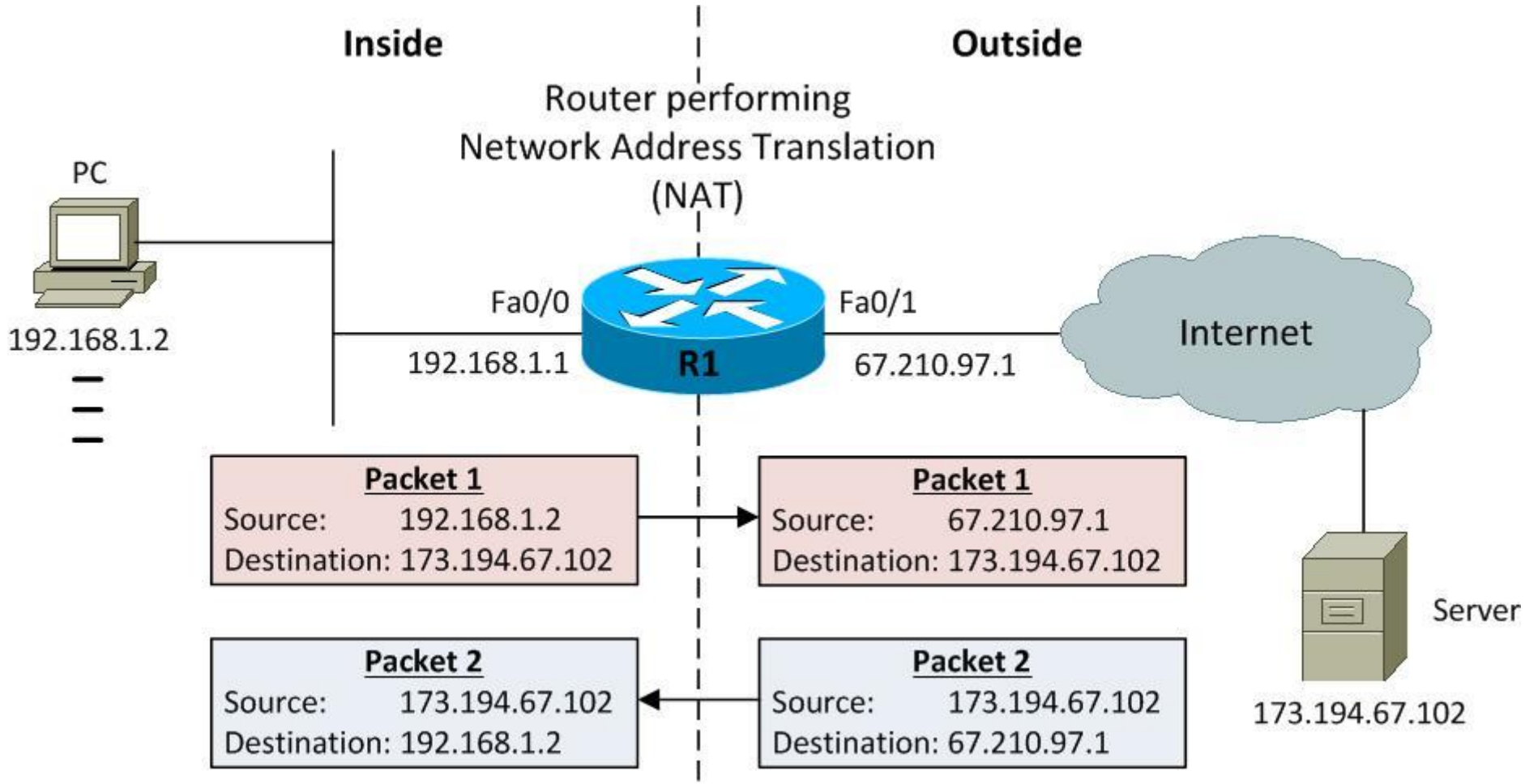
Problems Solved

- Can handle UDP query/response
- Can associate ICMP packets with connection
- Solves some of the inbound/outbound filtering issues — but state tables still need to be associated with inbound packets

Network Address Translators

- Translates source address (and sometimes port numbers)
- Primary purpose: coping with limited number of global IP addresses
- Sometimes marketed as a very strong firewall — is it?
- It’s not really stronger than a stateful packet filter

Basic NAT operation



https://www.freeccnastudyguide.com/study-guides/ccna/ch10/565-2/

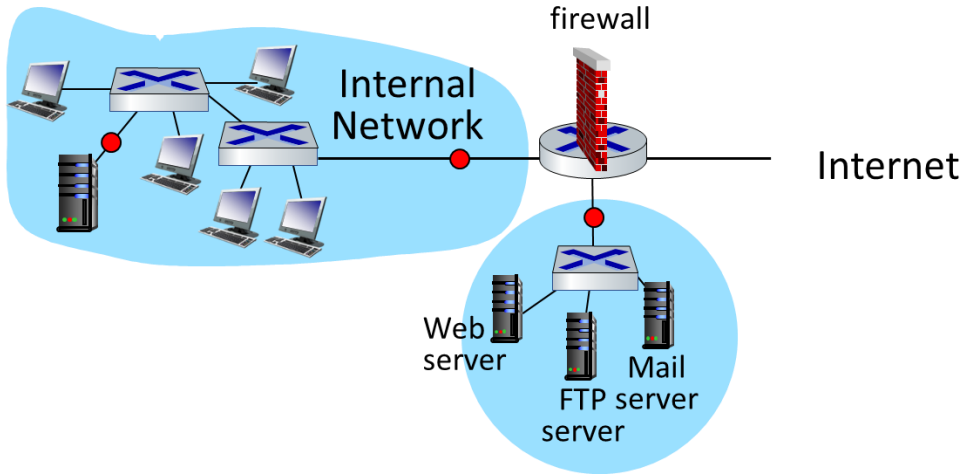
Comparison

- The lookup phase and the decision to pass or drop the packet are identical; all that changes is whether or not addresses are translated.

Stateful Packet Filter	NAT
Outbound Create state table entry.	**Outbound** Create state table entry. Translate address.
Inbound Look up state table entry; drop if not present	**Inbound** Look up state table entry; drop if not present. Translate address.

Types of Firewalls

- (Stateless) Packet Filters
- Stateful Packet Filters
- Application Gateways
- Circuit Relays
- Personal and/or Distributed Firewalls

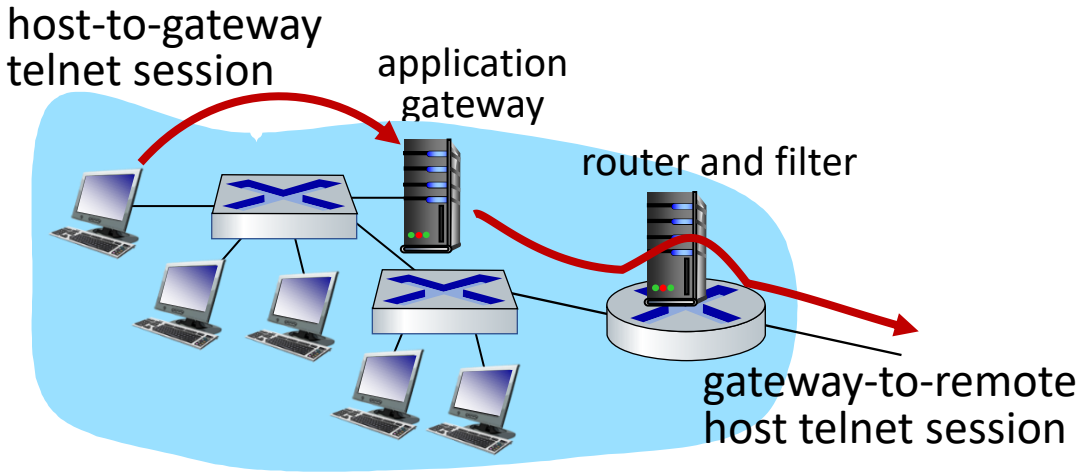


Moving Up the Stack

- Why move up the stack?
- Apart from the limitations of packet filters discussed last time, firewalls are inherently incapable of protecting against attacks on a higher layer
  - IP packet filters (plus port numbers. . .) can’t protect against bogus TCP data
  - A TCP-layer firewall can’t protect against bugs in SMTP
  - SMTP proxies can’t protect against problems in the email itself, etc.

Application gateways

- *example:* allow select internal users to telnet outside



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

Advantages

- Protection can be tuned to the individual application
- More context can be available
- You only pay the performance price for that application, not others

Disadvantages

- Application-layer firewalls **don't protect against attacks at lower layers!**
- They require a separate program per application
- These programs can be quite complex
- They may be very intrusive for user applications, user behavior, etc.

Example: Protecting Email

- Do we protect inbound or outbound email?
- Do we work at the SMTP level (RFC 2821) or the mail content level (RFC 2822)?
- What about MIME?
- What about encrypted email?
- What are the threats?

Email Threats

- The usual: defend against protocol implementation bugs
- Virus-scanning
- Anti-spam?
- Violations of organizational email policy?

Different Sublayers

- Note that are multiple layers of protection possible here
- The receiving machine can run a hardened SMTP, providing protection at that layer
- Once the email is received, it can be scanned at the content layer for any threats
- The firewall function can consist of either or both

Inbound Email

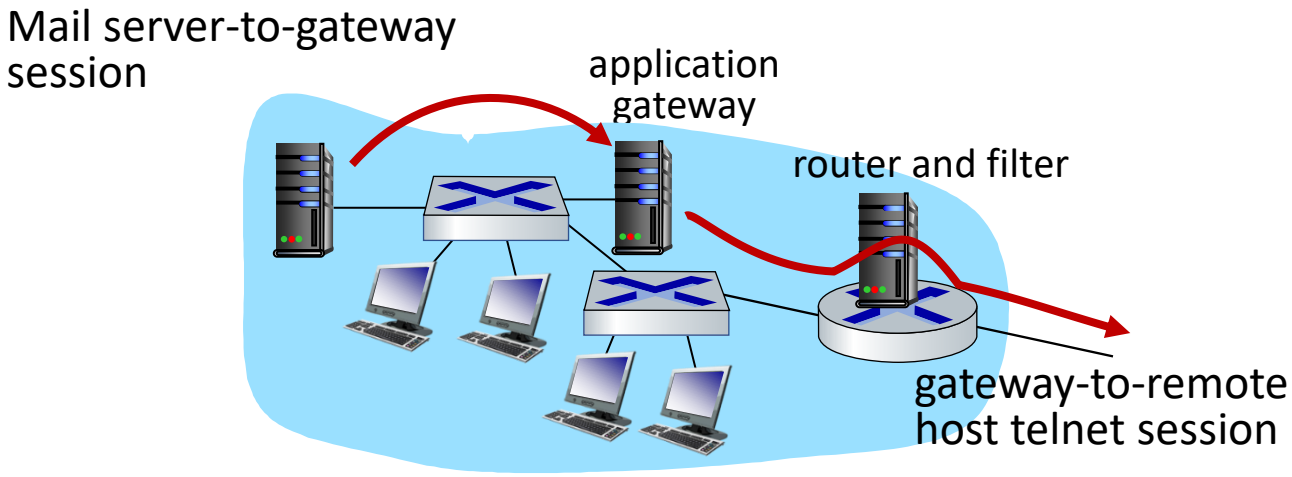
- Email is easy to intercept: MX records in the DNS route inbound email to an arbitrary machine
- Possible to use “*” to handle entire mail domain in local DNS server
- Net result: **all email for that domain is sent to a front end machine**

Outbound Email

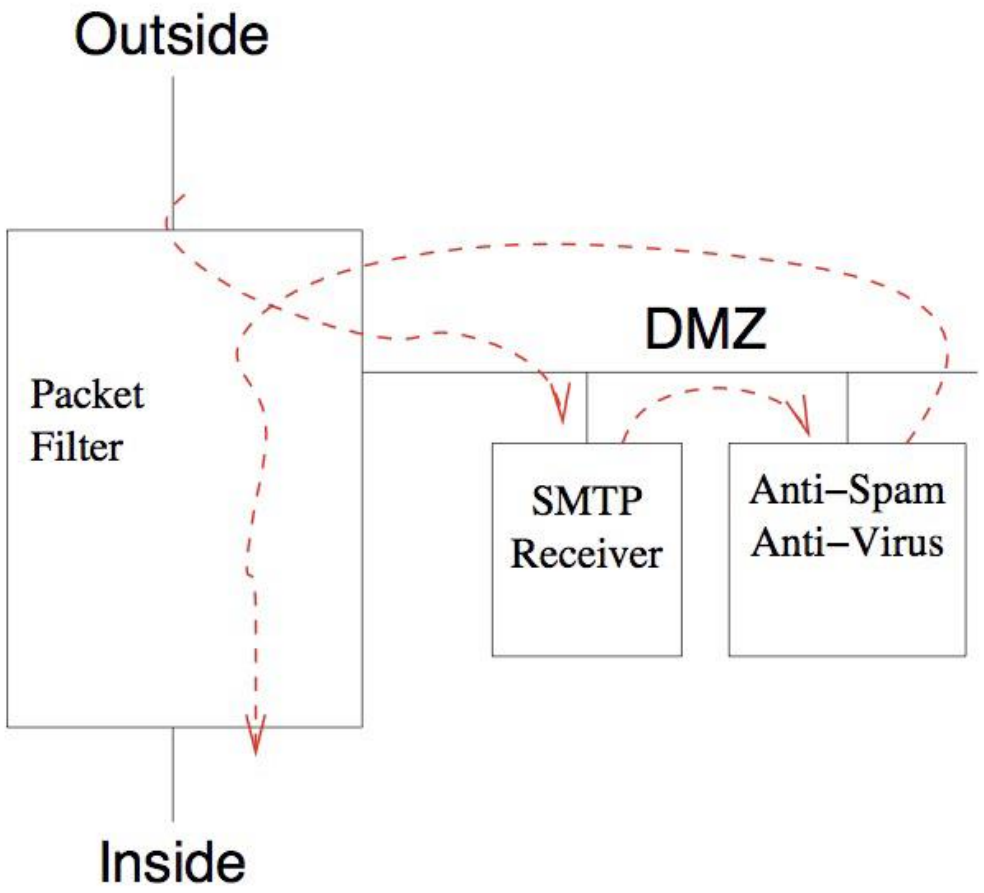
- **No help from the protocol** definition here
- But — most mailers have the ability to forward some or all email to a relay host
- Declare by administrator that this must be done
- Enforce this with a packet filter. . .

Combining Firewall Types

- Use an application firewall to handle inbound and outbound email
- Use a packet filter to enforce the rules



Firewalling Email



Enforcement

- Email can’t flow any other way
- The only SMTP server the outside can talk to is the SMTP receiver
- It forwards the email to the anti-virus/anti-spam filter, via some arbitrary protocol
- That machine speaks SMTP to some inside mail gateway
- Note the other benefit: if the SMTP receiver is compromised, it can’t speak directly to the inside

Outbound Email

- Again, we use a packet filter to block direct outbound connections to port 25
- The only machine that can speak to external SMTP receivers is the dedicated outbound email gateway
- That gateway can either live on the inside or on the DMZ

DNS: Internal Versus External View

- Should outsiders be able to see the names of all internal machines?
  - What about secretproject.foobar.com?
- Solution: use two DNS servers, one for internal requests and one for external request
- Put one on each side of the firewall

DNS Filtering

- All internal DNS queries go to a DNS switch
  - If it's an internal query, forward the query to the internal server or pass back internal NS record
  - If it's an external query, forward the query to outside, but:
    - Scrub the result to remove any references to inside machines
- Use a packet filter to block direct DNS communication

Small Application Gateways

- Some protocols don't need full-fledged handling at the application level
- That said, a packet filter isn't adequate
- Solution: examine some of the traffic via an application-specific proxy; react accordingly

FTP Proxy

- Remember the problem with the PORT command?
- Scan the FTP control channel
- If a PORT command is spotted, tell the firewall to open that port temporarily for an incoming connection

- Problem?

Attacks Via FTP Proxy

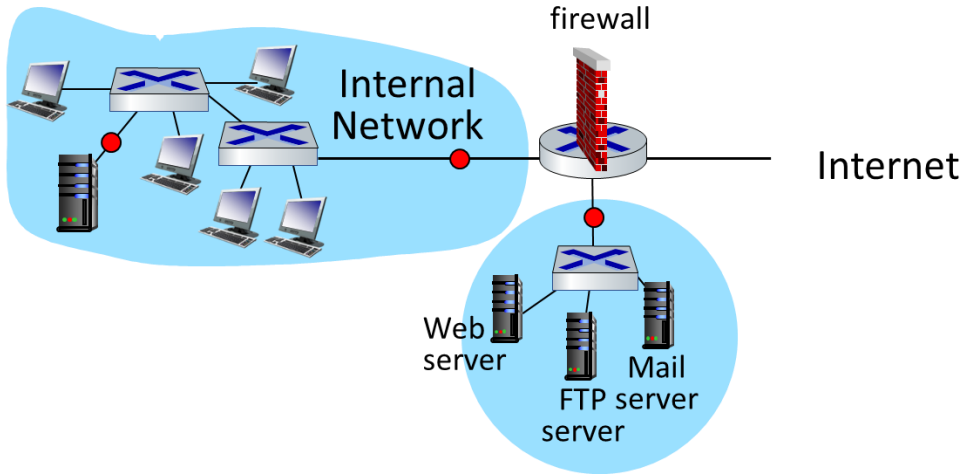
- Downloaded Java applets can call back to the originating host
- A malicious applet can open an FTP channel, and send a PORT command listing a vulnerable port on a nominally-protected host
- The firewall will let that connection through
- **Solution**: make the firewall smarter about what host and port numbers can appear in PORT commands. . .

Web Proxies

- Provide performance advantage: caching
- Can enforce site-specific filtering rules

Types of Firewalls

- (Stateless) Packet Filters
- Stateful Packet Filters
- Application Gateways
- Circuit Relays
- Personal and/or Distributed Firewalls

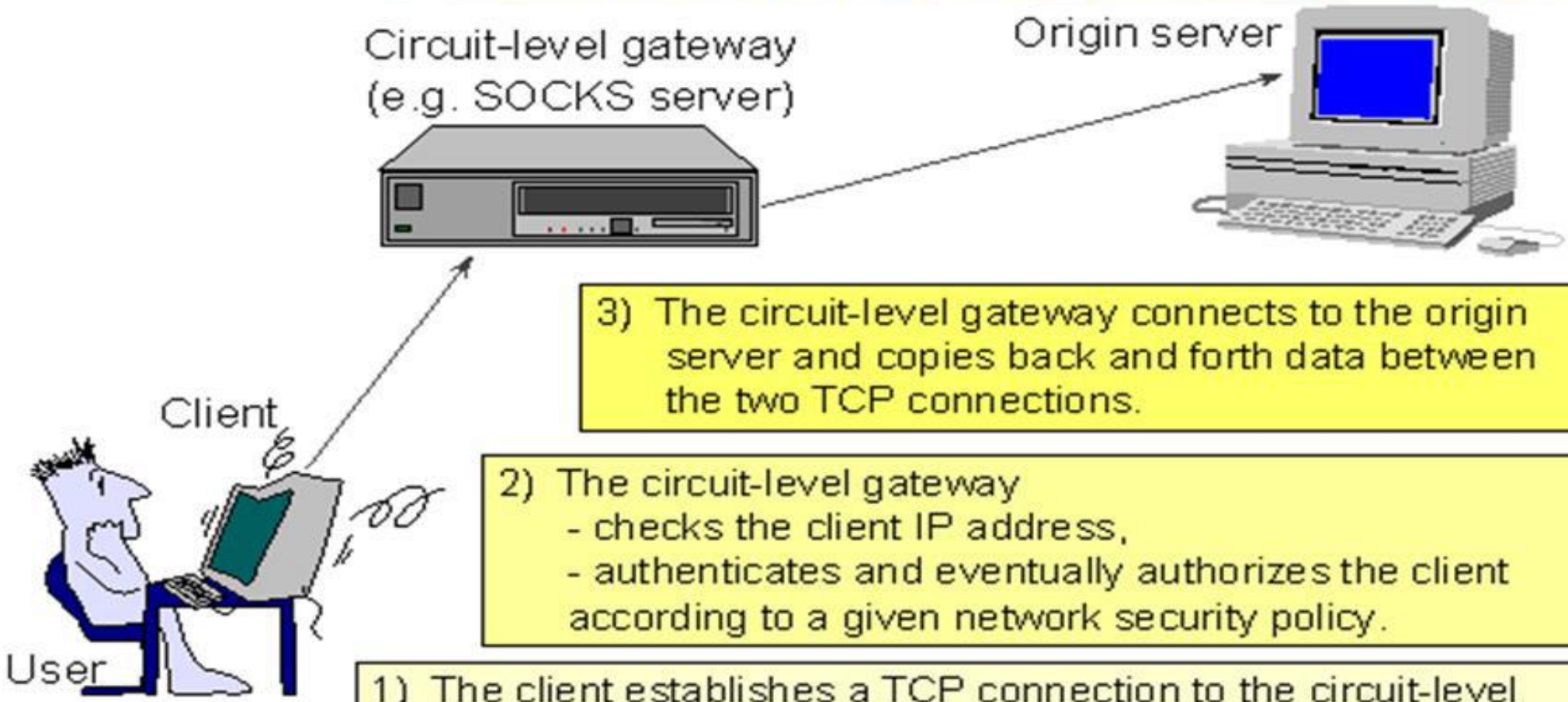


Circuit-level Gateway

- Sets up two TCP connections
- The gateway typically relays TCP segments from one connection to the other without examining the contents (no application-specific semantics)
- The security function consists of determining which connections will be allowed
- Typical use is a situation in which the system administrator does not have trust in the internal users
- Hides the IP address of internal machines (they never connect directly to external)
- Most common one: SOCKS
  - Supported by many common applications, such as Firefox and GAIM.

Proxy Servers and Firewalls

Circuit-level Gateways 11/19

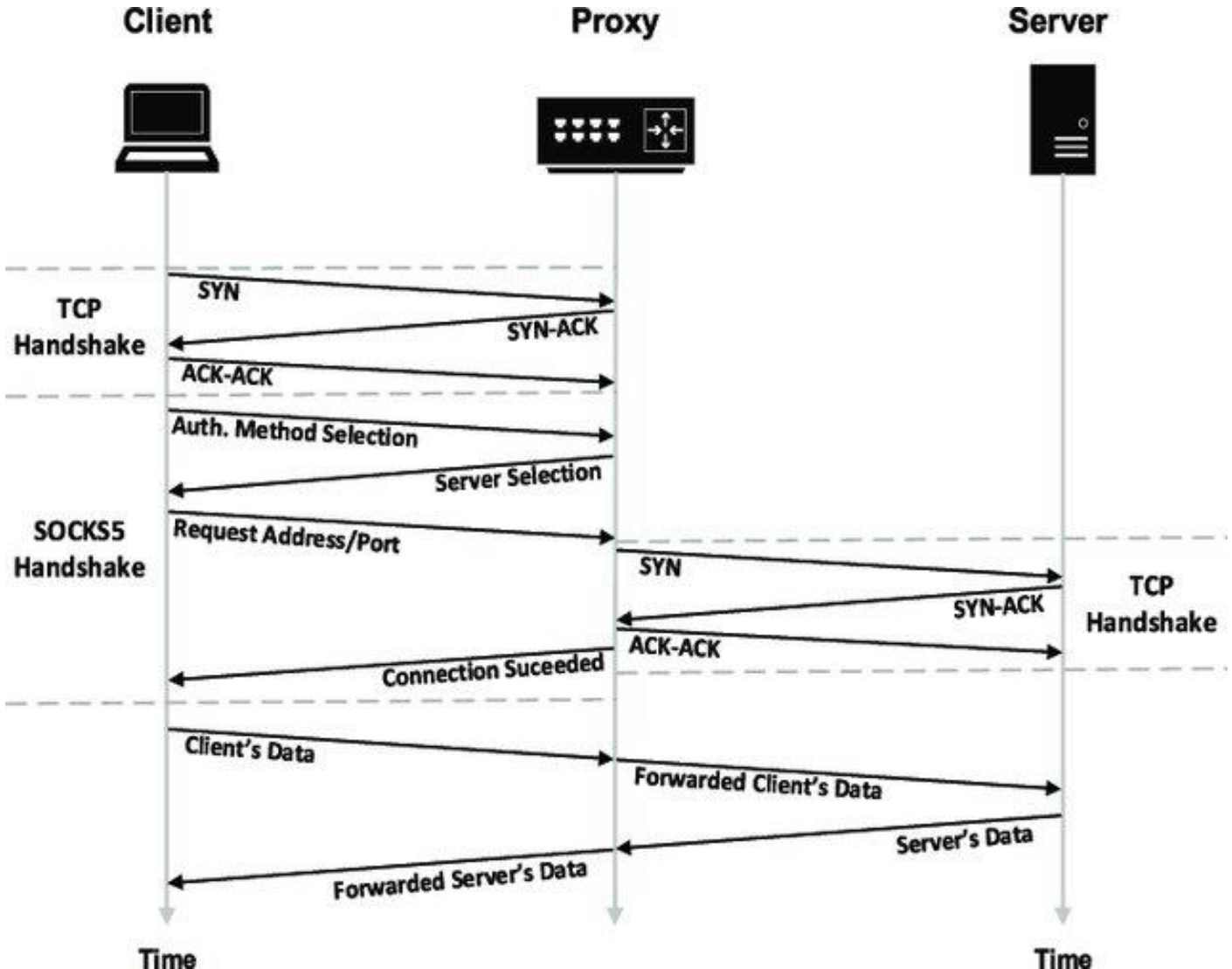


3) The circuit-level gateway connects to the origin server and copies back and forth data between the two TCP connections.

2) The circuit-level gateway
- checks the client IP address,
- authenticates and eventually authorizes the client according to a given network security policy.

1) The client establishes a TCP connection to the circuit-level gateway and requests a second TCP connection to a remote server (origin server)

SOCKS proxy protocol



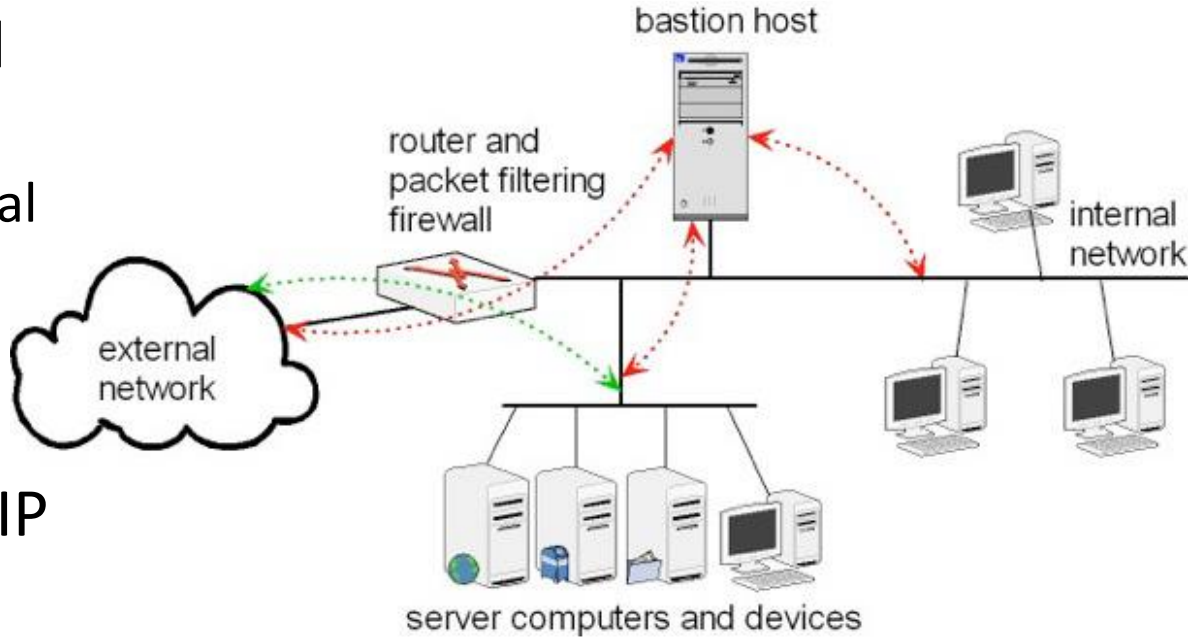
Firewall Configurations

Bastion Host

- A system identified by the firewall administrator as a critical strong point in the network´s security
- The bastion host serves as a platform for an application-level or circuit-level gateway
- The bastion host performs authentication and proxy functions
- May be single or multi-homed

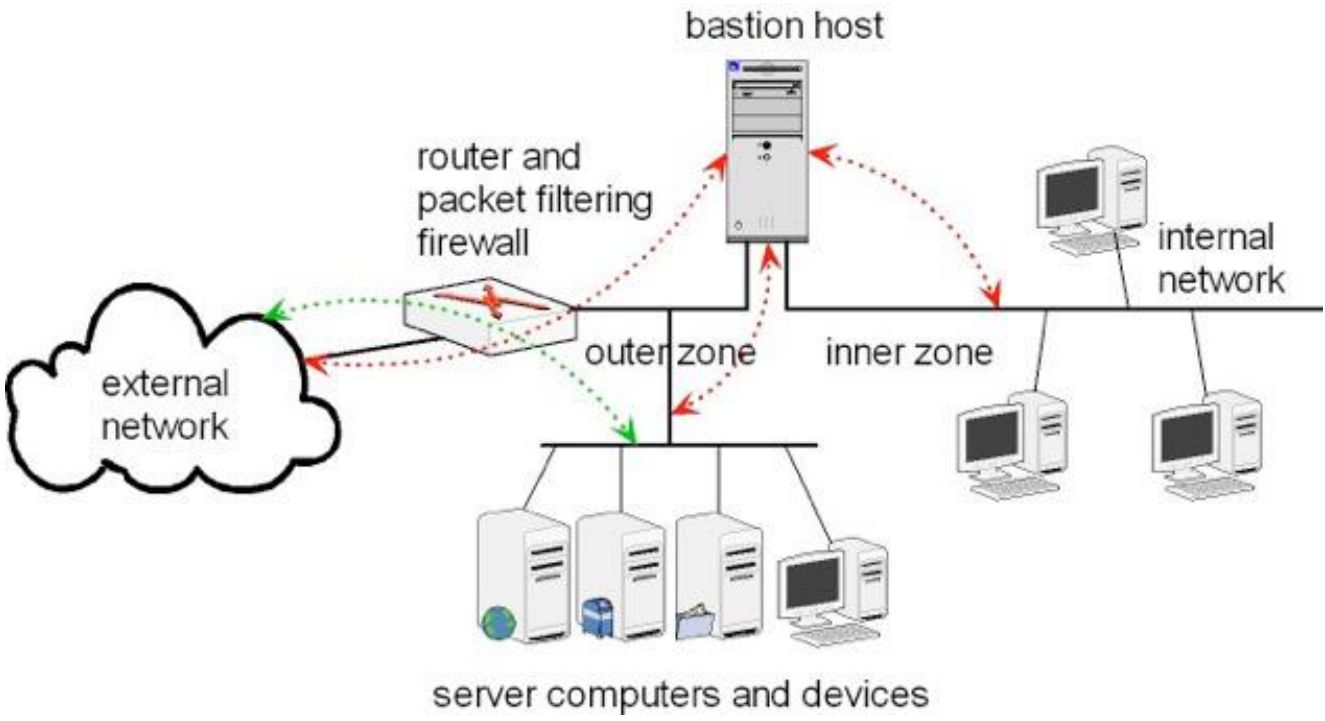
Single-Homed Bastion System

- Consists of a packet-filtering router and a bastion host
  - Router connects internal network to external network
  - Bastion host is inside the internal network
- PF firewall inspects each egress and blocks it if its source address is not the IP address of bastion host
- If the PF router is compromised, the attacker can modify the ACLs and bypass the bastion host



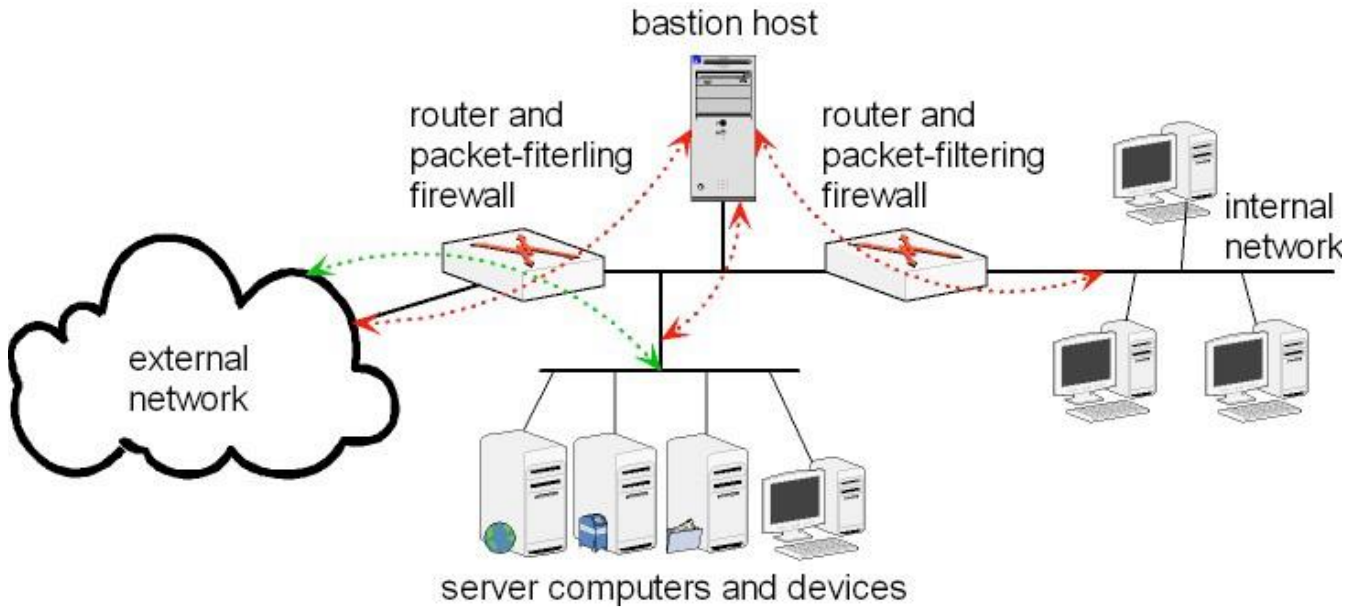
Dual-Homed Bastion System

- Two zones (homes) in the internal network:
  - Inner zone: hosts are unreachable from external
  - Outer zone: hosts may be reached from Internet
- Hosts in inner zone are protected by both bastion host and PF router
- Servers in outer zone protected by PF router
- Prevents access to the internal network even if the PF router is compromised
  - Point of failure



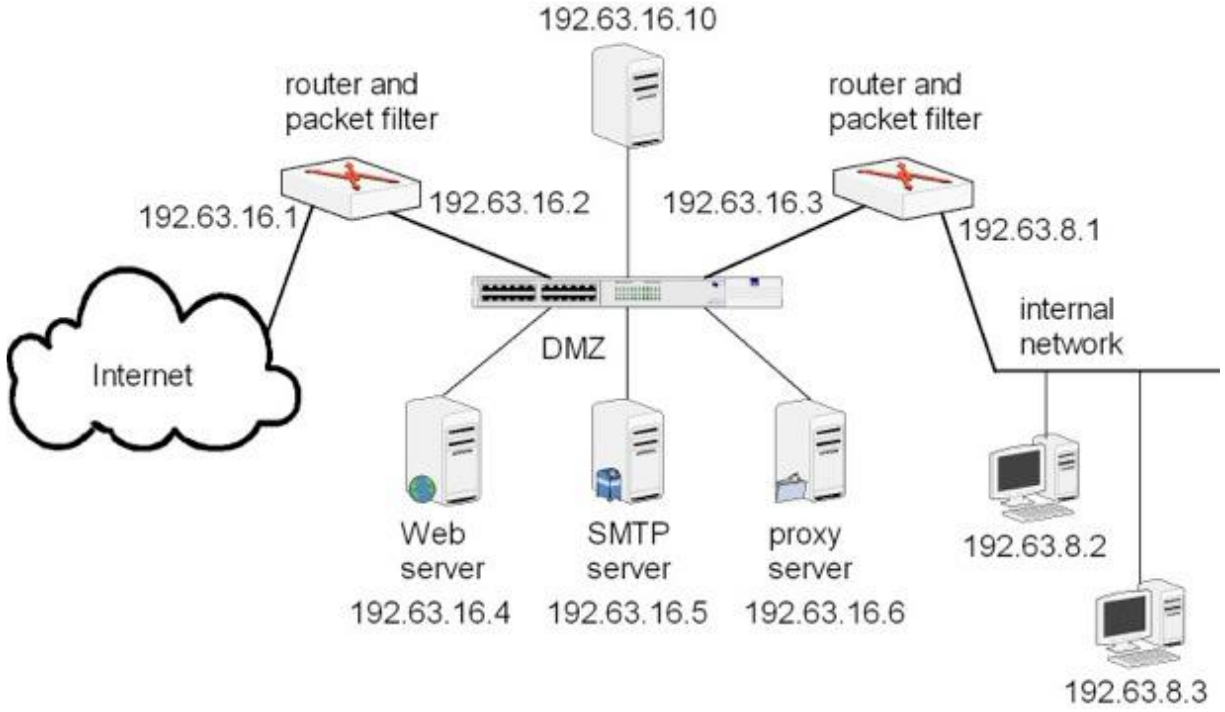
Screened Subnets

- A SHBH network paired with a second PF router for the internal network
- Area between the two PF routers is called a screened subnet
- Advantages
  - Three levels of defense
  - Hides the internal network structure from external hosts
  - The outside router advertises only the existence of the screened subnet to the internet
  - The inside router advertises only the existence of the screened subnet to the internal network



Demilitarized Zones (DMZ)

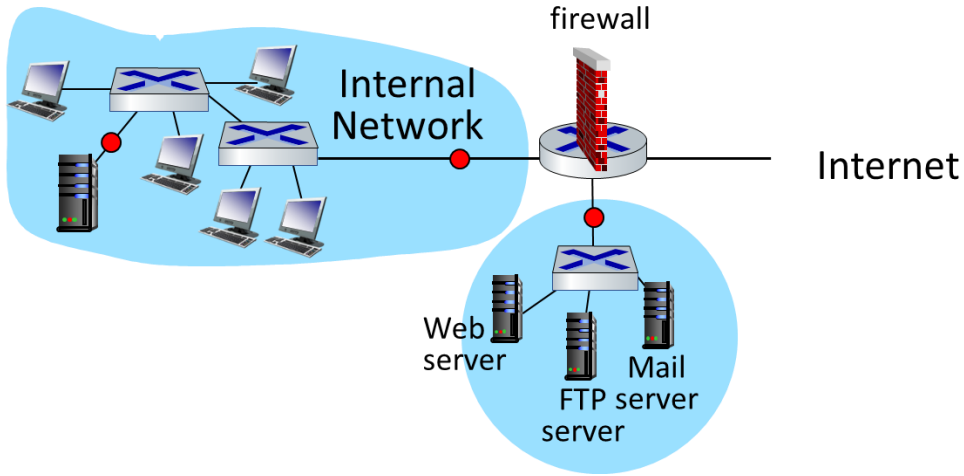
- A subnet between two firewalls in an internal network
  - External firewall protects DMZ from external threats
  - Internal firewall protects internal network from DMZ



- Performance? Price?
  - FTP server behind Firewall

Types of Firewalls

- (Stateless) Packet Filters
- Stateful Packet Filters
- Application Gateways
- Circuit Relays
- Personal and/or Distributed Firewalls



Rationale

- Conventional firewalls rely on topological assumptions — these are questionable today
  - Laptops, more or less by definition, travel
  - When they're outside the firewall, what protects them?
- Instead, install protection on the end system
- Let it protect itself

Personal Firewalls

- Add-on to the main protocol stack
- The “inside” is the host itself; everything else is the “outside”
- Most act like packet filters
- Rule set can be set by individual or by administrator

Saying “No”, Saying “Yes”

- It’s easy to reject protocols you don’t like with a personal firewall
- The hard part is saying “yes” safely
- There’s no topology — all that you have is the sender’s IP address

Application-Linked Firewalls

- Most personal firewalls act on port numbers
- Some firewalls are tied to applications
  - individual programs are or are not allowed to talk, locally or globally
- Pros: don't worry about cryptic port numbers; handle auxiliary ports just fine
- Cons: application names can be just as cryptic; service applications operate on behalf of some other application

Distributed Firewalls

- In some sense similar to personal firewalls, though with central policy control
- Use IPsec to distinguish “inside” from “outside”
  - Insiders have inside-issued certificates; outsiders don’t
  - Only trust other machines with the proper certificate
- No reliance on topology; insider laptops are protected when traveling; outsider laptops aren’t a threat when they visit

The Problems with Firewalls

Corrupt Insiders

- Firewalls assume that everyone on the inside is good!!!
  - Obviously, that's not true
  - Beyond that, active content and subverted machines mean there are bad actors on the inside

Connectivity

- Firewalls rely on topology
- If there are too many connections, some will bypass the firewall
  - Sometimes, that's even necessary; it isn't possible to effectively firewall all external partners
  - A large company may have hundreds or even thousands of external links, most of which are unknown to the official networking people

Laptops

- Laptops, more or less by definition, travel
- When they're outside the firewall, what protects them?
  - At one conference, it was seen than at least a dozen other attendee machines were infected with the Code Red virus
    - (Code Red only infected web servers. Why were laptops running web servers?)

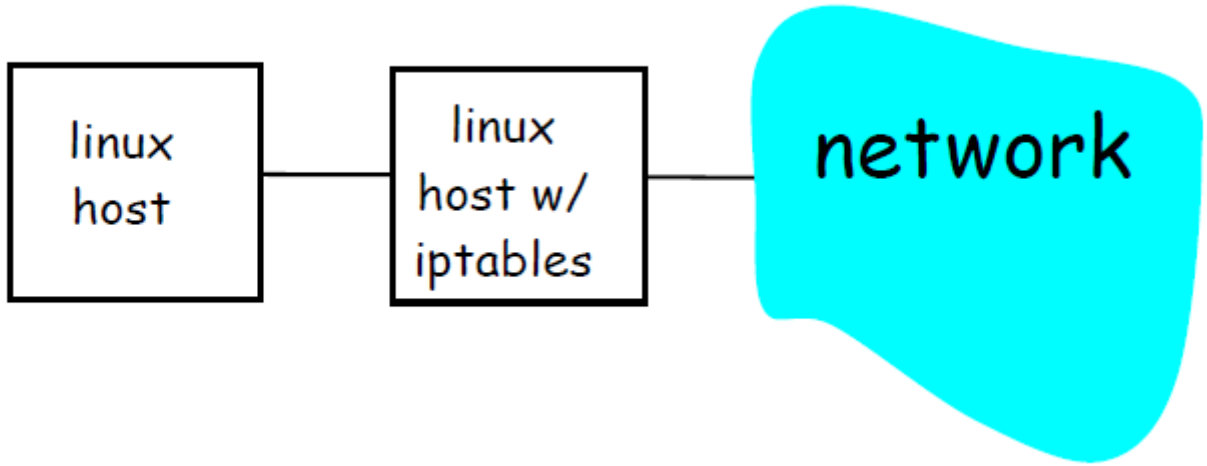
Evasion

- Firewalls and firewall administrators got too good
- Some applications weren't able to run
- Vendors started building things that ran over HTTP
  - HTTP usually gets through firewalls and even web proxies. . .

iptables

iptables deployment

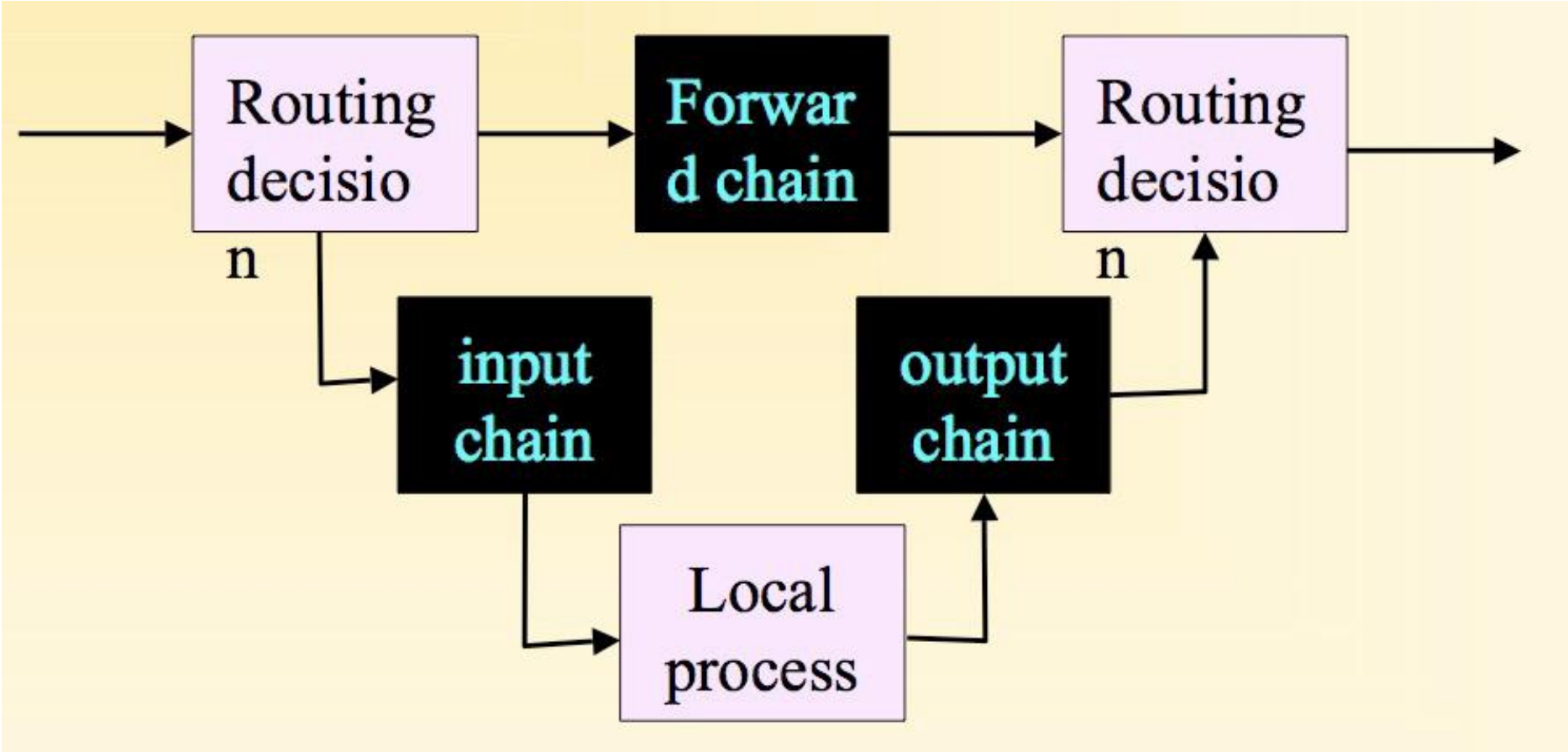
- Converts Linux box into a packet filter.
- Included in most Linux distributions today.



Rule Chains

- To organize groups of rules, many firewalls allow you to define "chains" that consist of a set of rules in a particular order.
  - The chains can be included in other 'chains' leading to a hierarchical arrangement.
  - Typically there are default chains corresponding to the 3 core paths of packets:
    - INPUT, OUTPUT, FORWARD
- Chains allow coherent sets of rules to be grouped and shared.
  - For example the rules for a FTP service could be grouped into a chain and then used on several different firewalls to apply the same policy to each.

Iptable filter chain structure



iptables: Example command

```
iptables -A INPUT -i eth0 -s 232.16.4.0/24 -j ACCEPT
```

- Sets a rule
  - Accepts packets that enter from interface eth0 and have source address in 232.16.4/24
- Kernel applies the rules in order.
  - The first rule that matches packet determines the action for that packet
- Append: -A
  - Adds rule to bottom of list of existing rules

iptables: More examples

```
iptables -L
```

- list current rules

```
iptables -D INPUT 2
```

- deletes 2nd rule in INPUT chain

```
iptables -I INPUT 1 -p tcp -tcp-flags SYN -s
  232.16.4.0/24 -d 0/0:22 -j ACCEPT
```

- -I INPUT 1, put rule at top
- Accept TCP SYNs to port 22 (ssh) from 232.16.4.0/24

Limitations of firewalls, gateways

- IP spoofing: router can't know if data "really" comes from claimed source
- if multiple apps 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

- filters often use all or nothing policy for UDP
- *tradeoff:* degree of communication with outside world, level of security
- many highly protected sites still suffer from attacks

References

- Kurose, James F., and Keith W. Ross. "Computer networking: A top-down approach edition." Addision Wesley (2007), chapter 8.
- Steven Bellovin, COMS W4180, Columbia University, 2006
- Mehdi Kharrazi, CE40-817, Sharif University of Technology, 2015
- Vitaly Shmatikov, CS 361S, UT Austin, 2014