CE 817 - Advanced Network Security

Lecture 4

Mehdi Kharrazi
Department of Computer Engineering
Sharif University of Technology

Acknowledgments: Some of the slides are fully or partially obtained from other sources. Reference is noted on the bottom of each slide, when the content is fully obtained from another source. Otherwise a full list of references is provided on the last slide.

Types of Firewalls

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

- Many firewalls are combinations of these types.

[Bellovin 06]

Application 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.

[Bellovin06]

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

[Bellovin06]

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.

[Bellovin06]

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?

[Bellovin06]

Email Threats

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

[Bellovin06]

Different Sublayers

- Note that are 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

[Bellovin06]

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 domain
- Net result: all email for that domain is sent to a front end machine

[Bellovin06]

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. . .

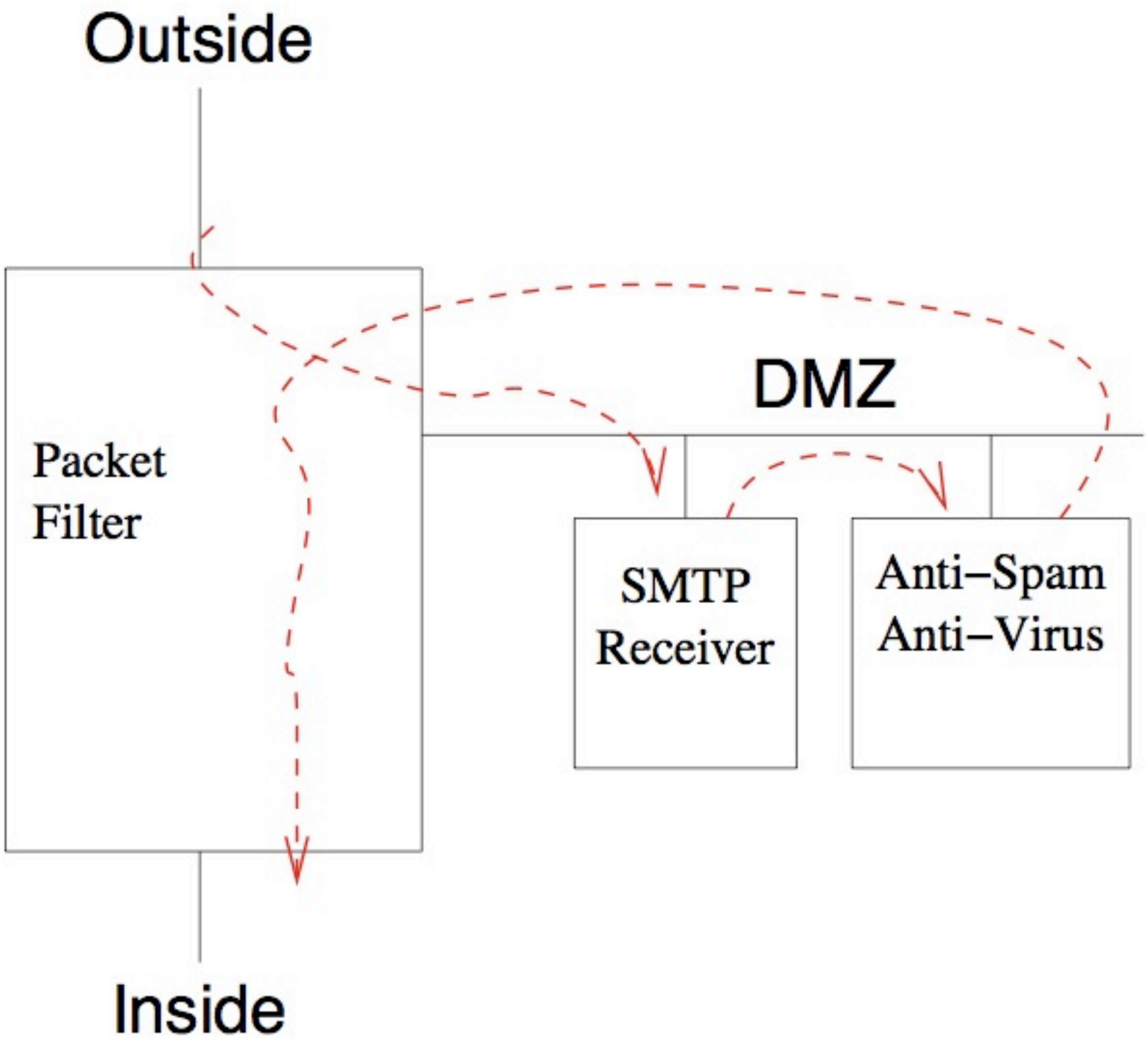
[Bellovin06]

Combining Firewall Types

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

[Bellovin06]

Firewalling Email



[Bellovin06]

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

[Bellovin06]

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

[Bellovin06]

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

[Bellovin06]

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

[Bellovin06]

Application Proxies

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

[Bellovin06]

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

[Bellovin06]

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. . .

[Bellovin06]

Web Proxies

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

[Bellovin06]

Circuit Gateways

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 trusts 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.

[Stanton08,Bellovin06]

Application Modifications

- Application must be changed to speak the circuit gateway protocol instead of TCP or UDP
- Easy for open source
- Socket-compatible circuit gateway libraries have been written for SOCKS — use those instead of standard C library to convert application

[Bellovin06]

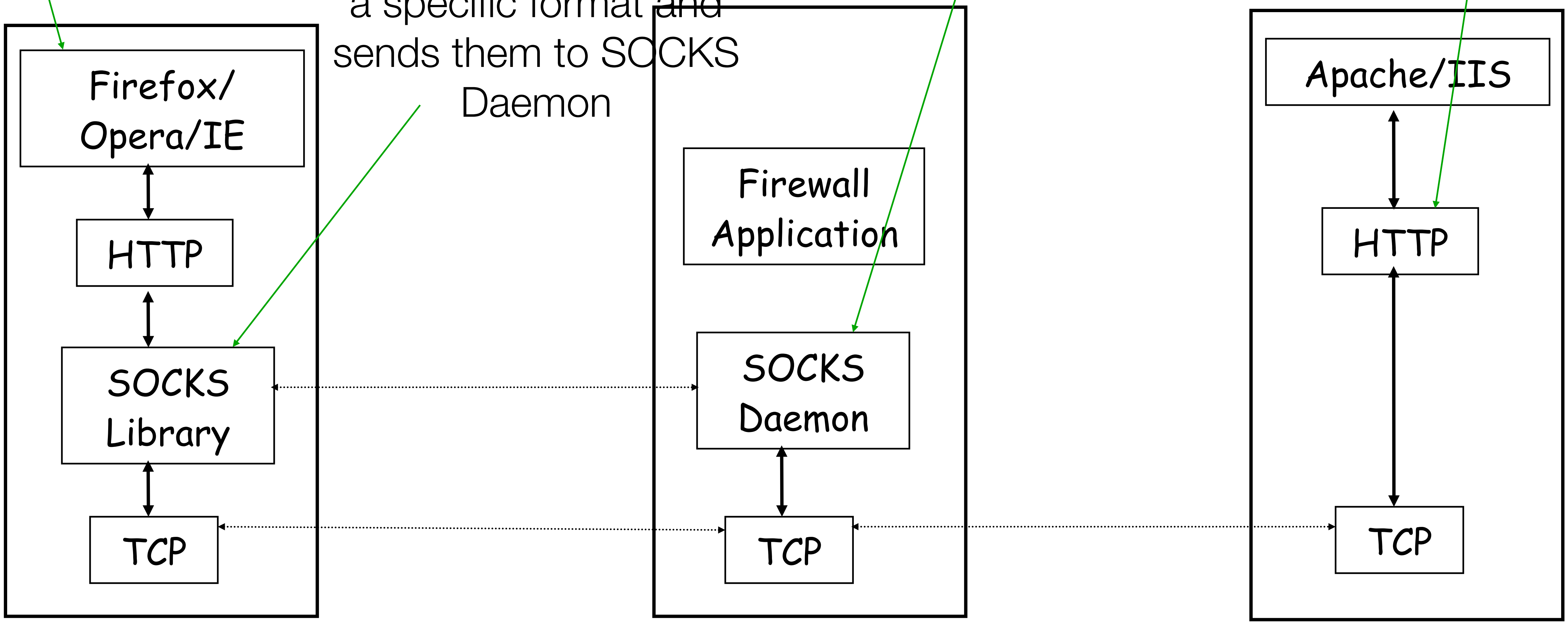
SOCKS proxy protocol

1. For example, let's assume that browser requests a page

2. SOCKS Library is a collection of procedures. It translates requests into a specific format and sends them to SOCKS Daemon

3. The SOCKS Daemon runs on the firewall host. It is independent of firewall software. The daemon authenticates the user and forwards all the data to the server.

4. The server receives requests as ordinary HTTP. It does not need a SOCKS library.

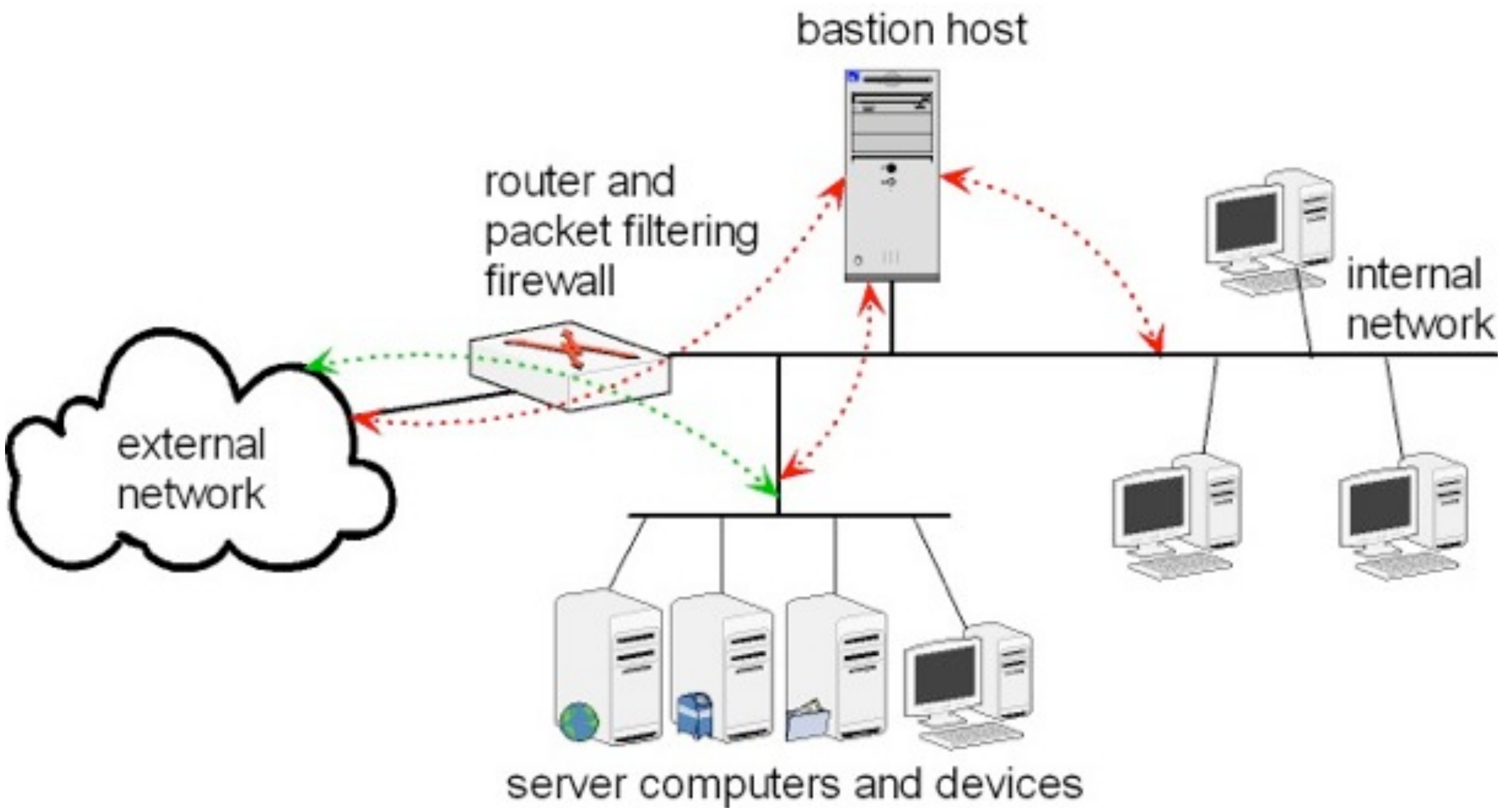


[Ross]

Firewall Configurations

Single-Homed Bastion System

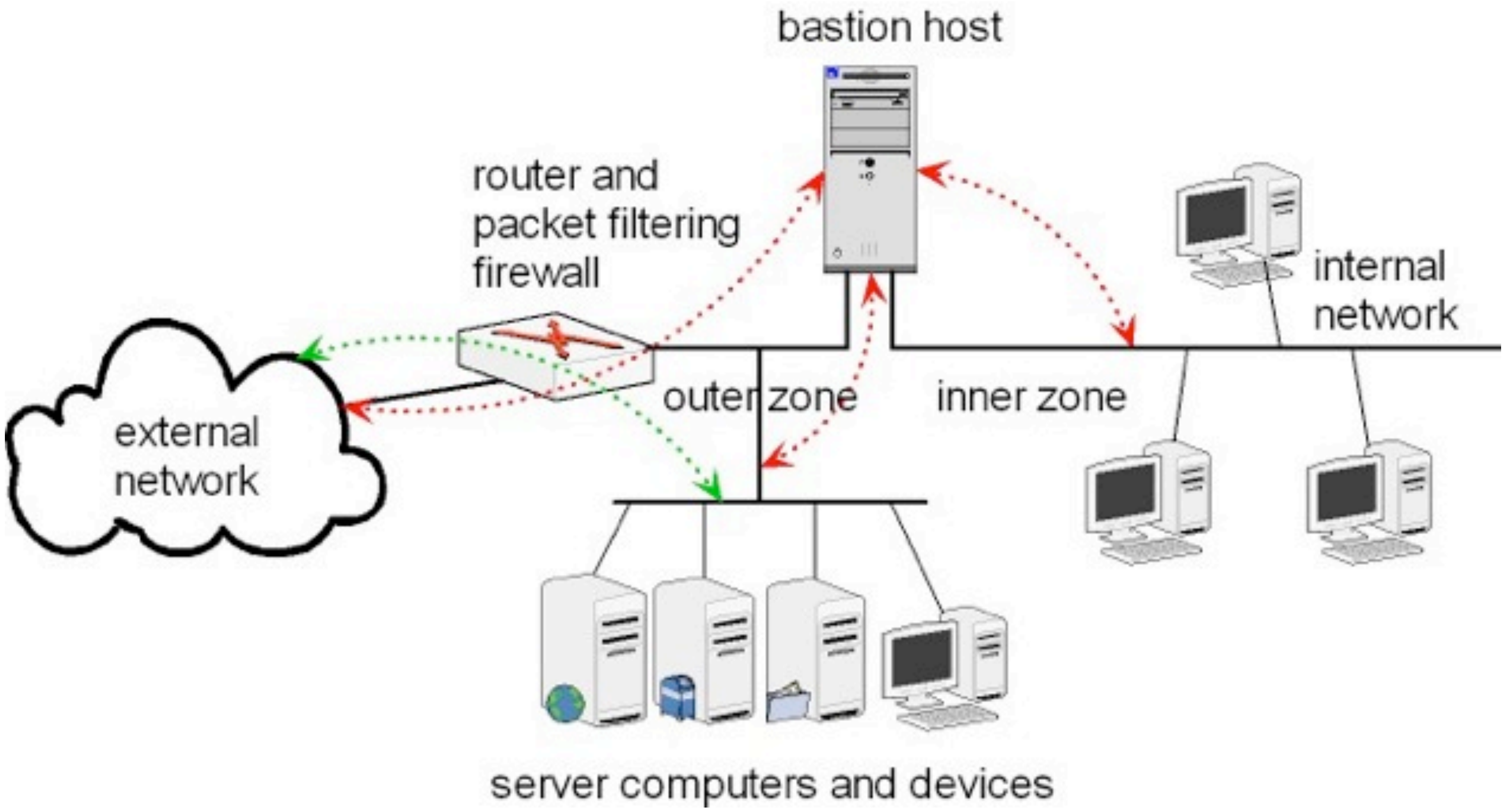
- Consists of a packet-filtering router and a bastion host
  - Router connects internal network to external network
  - Bastion host is inside the internal network
- PF firewall inspects each egress and blocks it if its source address is not the IP address of bastion host
- If the PF router is compromised,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


bastion host
router and packet filtering firewall
internal network
external network
server computers and devices

J. Wang. Computer Network Security Theory and Practice. Springer 2008

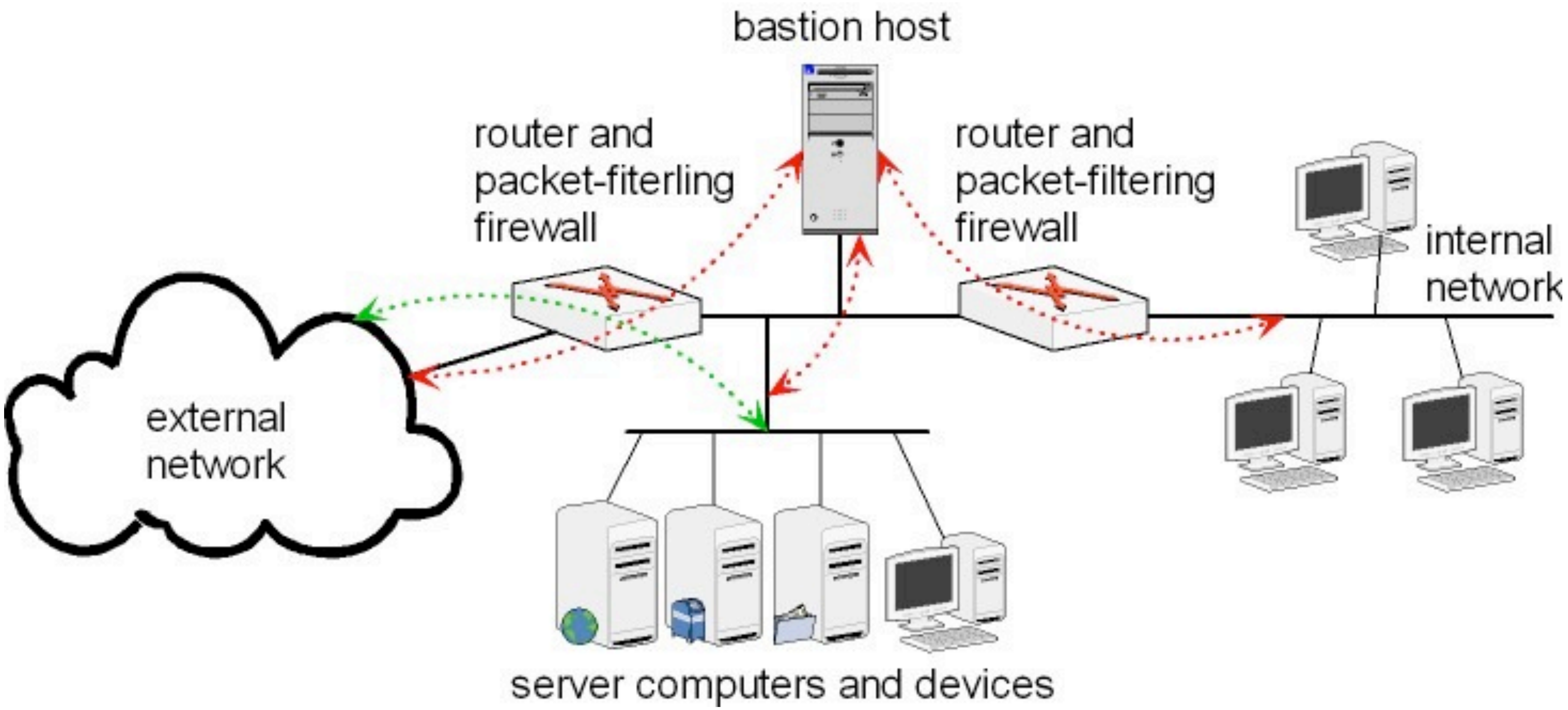
Dual-Homed Bastion System

- Two zones 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



J. Wang. Computer Network Security Theory and Practice. Springer 2008

Screened Subnets

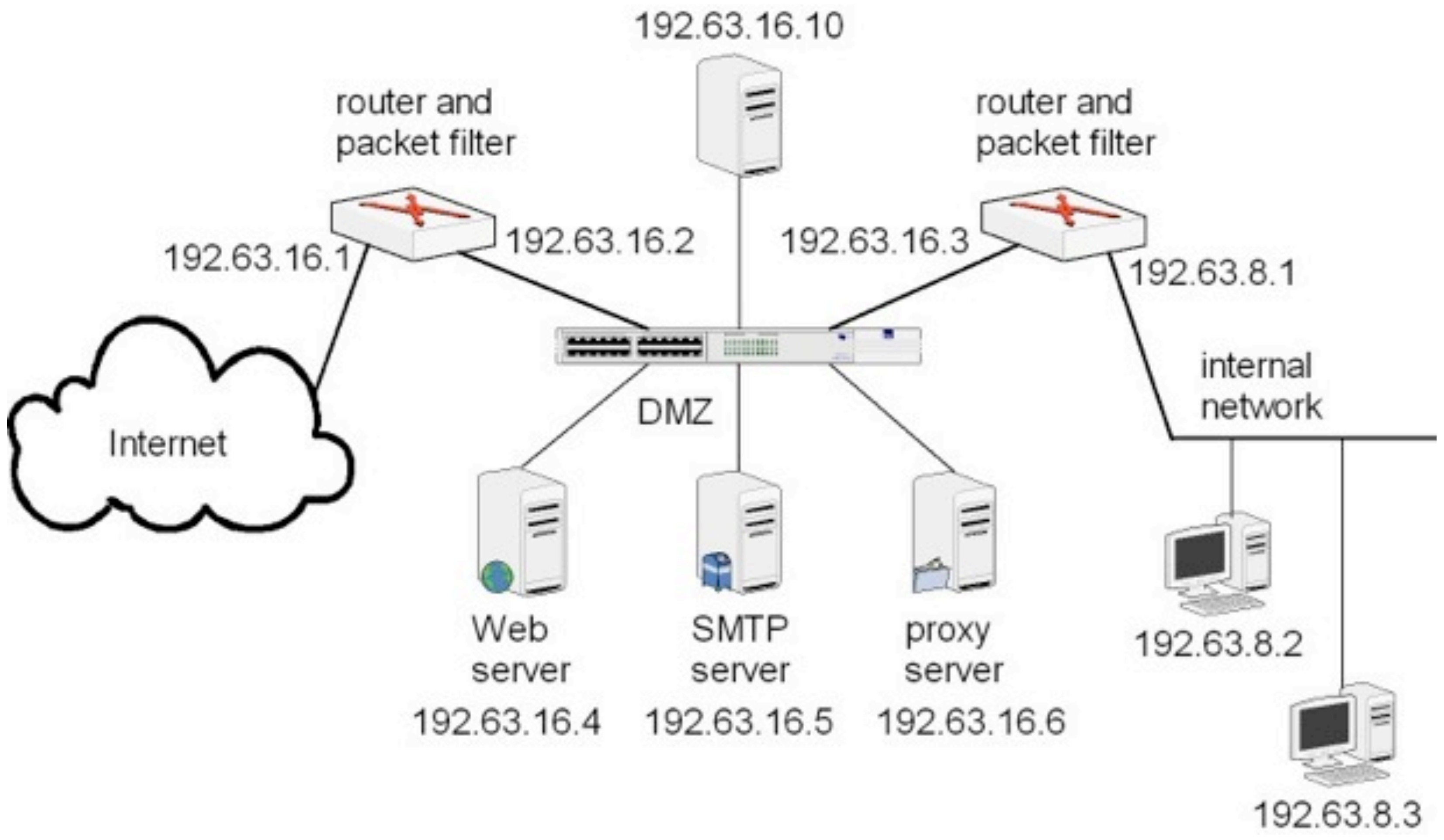


- A SHBH network paired with a second PF router for the internal network
- Area between the two PF routers is called a screened subnet
- Hides the internal network structure from external hosts

J. Wang. Computer Network Security Theory and Practice. Springer 2008

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



J. Wang. Computer Network Security Theory and Practice. Springer 2008

Personal and Distributed Firewalls

Rationale

- Conventional firewalls rely on topological assumptions — these are questionable today
- Instead, install protection on the end system
- Let it protect itself

[Bellovin06]

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

[Bellovin06]

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

[Bellovin06]

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

[Bellovin06]

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

[Bellovin06]

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

[Bellovin06]

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

[Bellovin06]

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?)

[Bellovin06]

Evasion

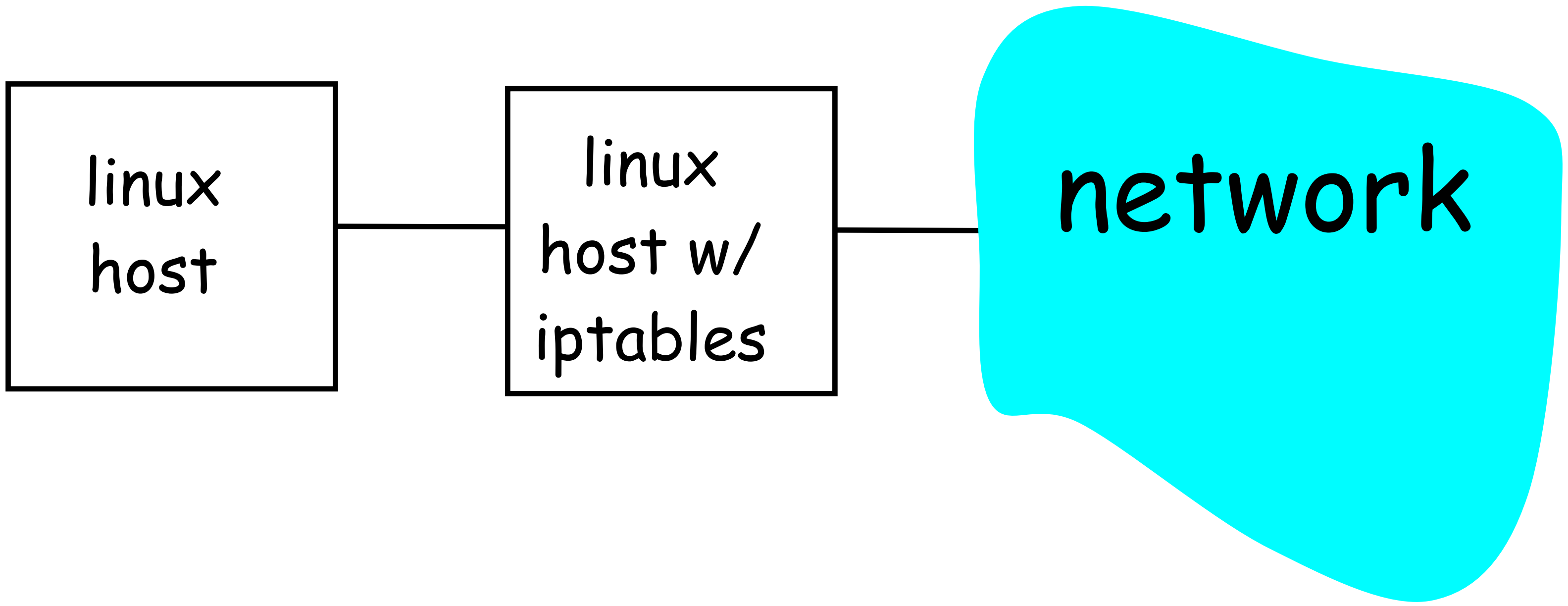
- Firewalls and firewall administrators got too good
- Some applications weren’t able to run
- Vendors started building things that ran over HTTP
- HTTP usually gets through firewalls and even web proxies. . .

[Bellovin06]

iptables

iptables deployment

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



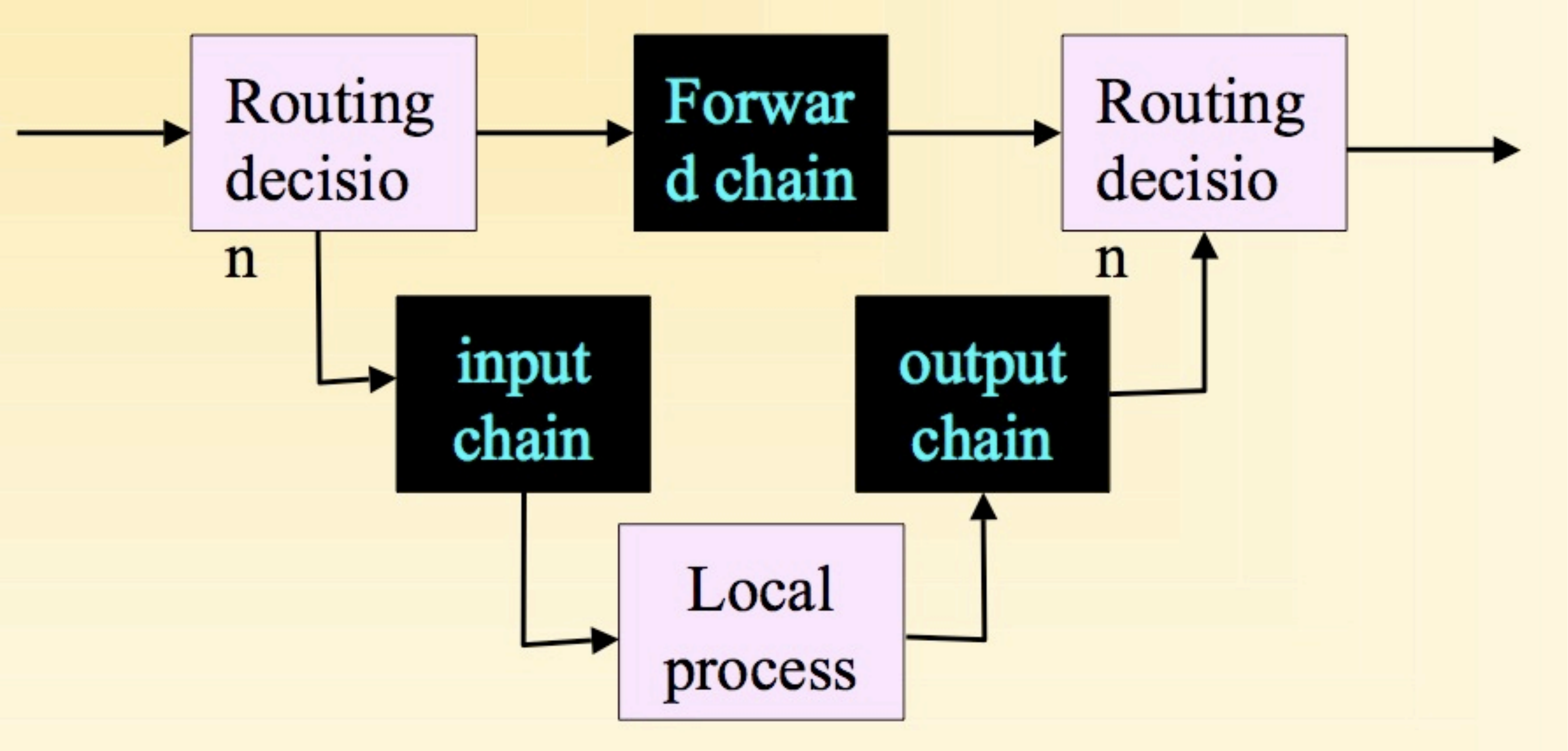
[Ross07]

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.

[Stanton 08]

Iptable filter chain structure



[Stanton 08]

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

[Ross07]

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

[Ross07]

Acknowledgments/References

- [Bellovin06] COMS W4180 — Network Security Class Columbia University, Steven Bellovin, 2006.
- [Stanton08] Network Security, CS 192/286, George Washington University, Jonathan Stanton, 2008.
- [Ross07] Network Security, CS 393/682, Spring 2007. Keith Ross. Polytechnic University.