



# Firewalls

David Parter


University of Wisconsin  
Computer Sciences Department  
Computer Systems Lab  
dparter@cs.wisc.edu

November 11, 2010



# Topics

- Firewall basics
- Types of Firewalls
- Deployment scenarios
- Related Technologies
- Real World Experience
- Summary
- Questions



# Firewall Basics

- Security model
- Types of firewalls
- Firewall rules

# Security Model

- Perimeter security
  - Like a guard at the gate, checking ID badges
  - Assumes that “inside” is trusted, “outside” is not
  - Larger area inside perimeter -> more complexity, weaker security
  - Smaller perimeter -> more specific security
- Applies predefined access rules



# Why Use a Firewall?

- Protect vulnerable services
  - Poorly designed protocols
  - Poorly implemented protocols/services
- Protect vulnerable computers/devices
  - Poorly configured
  - Can't be configured
  - Can't be patched

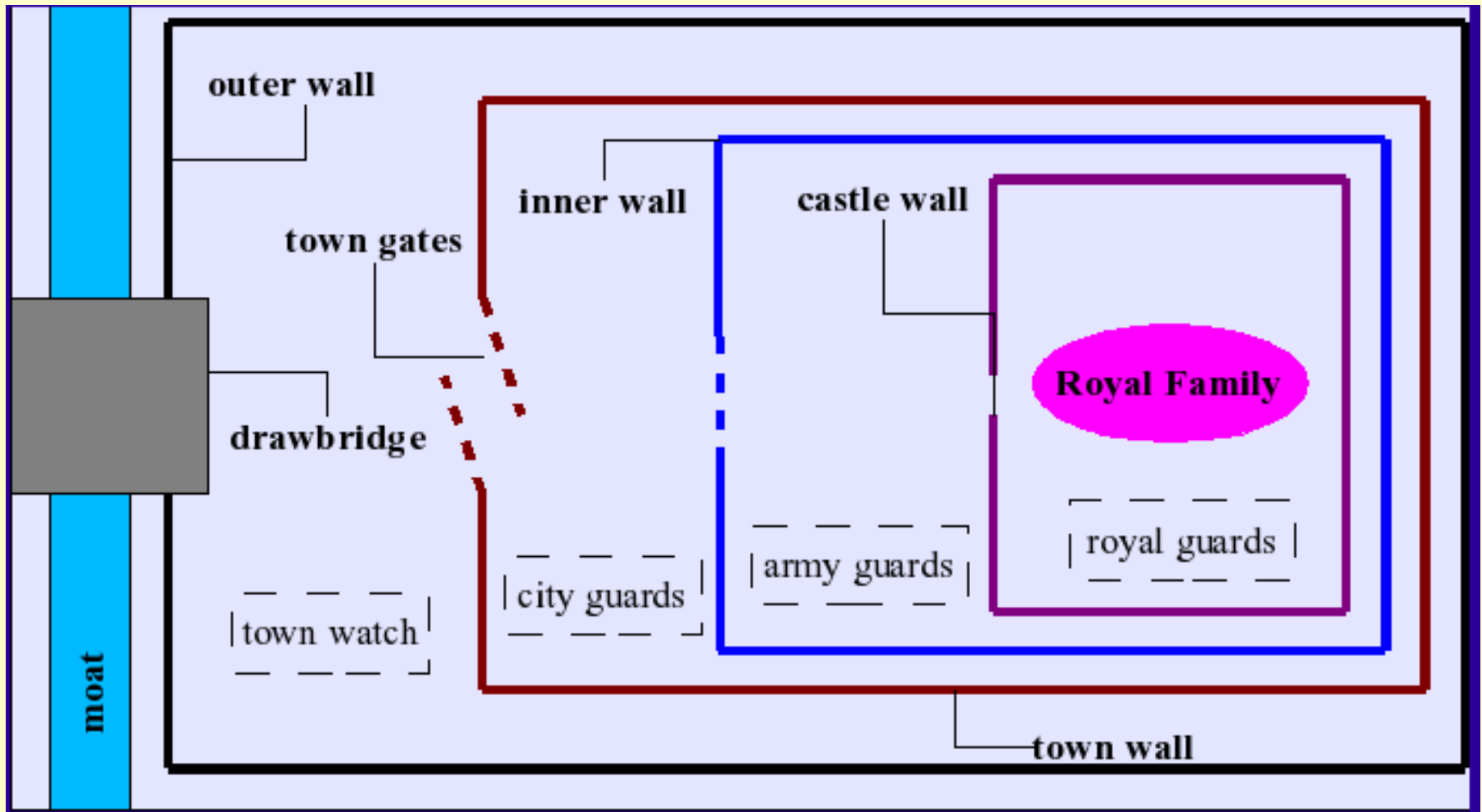
# Why Use a Firewall?

- To protect an “appliance”
- Protect a system that can not be upgraded
  - Version/upgrade restrictions from vendor
  - ex: printers; data acquisition devices; scientific “instruments”; devices with customized & embedded versions of popular operating systems; devices with embedded web servers for configuration or control ...

# Why Use a Firewall?

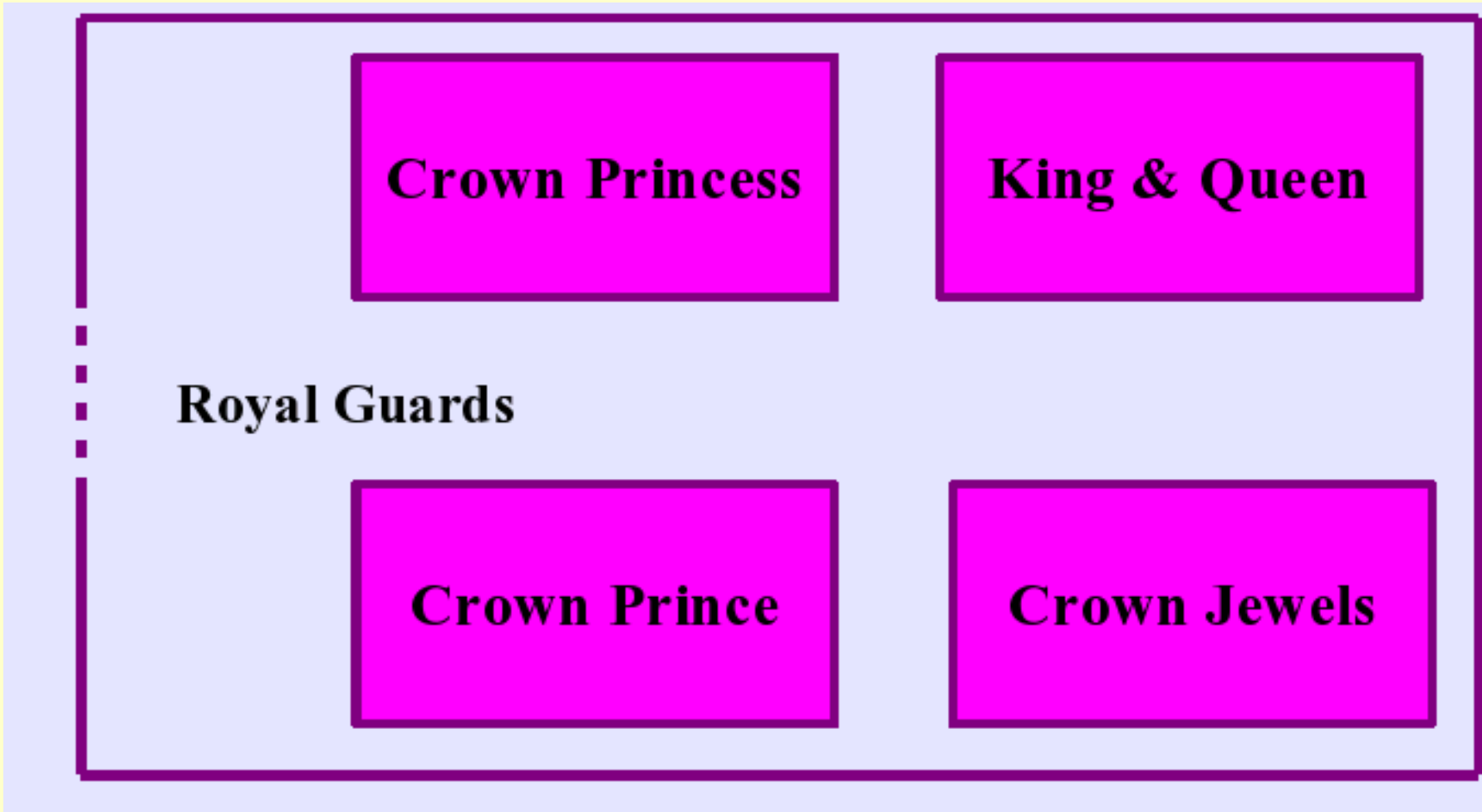
- Defeat some denial of service (DOS) attacks
  - If the firewall has enough bandwidth
- Considered an “easy” solution
  - Satisfy “check-box” requirements
  - Only need to deal with security in one place (not really an advantage from a total security point of view)

# Perimeter Security and Defense in Depth





# Improved Security: Reduced Perimeters



# Types of Firewalls

- Basic Technology Options:
  - Packet Filtering (screening)
  - Application Proxy
- Other Factors:
  - State full vs. Stateless
  - Router vs. Bridge
  - Configuration/Security model

# Packet Filtering

- Acts like a router or bridge
  - Does not modify network connections or packet headers
- Allow/Deny packets based on packet data
  - Source, destination, port, etc
- Allow/Deny packets based on Input/Output interfaces
  - physical or logical network topology



# Filter on packet data

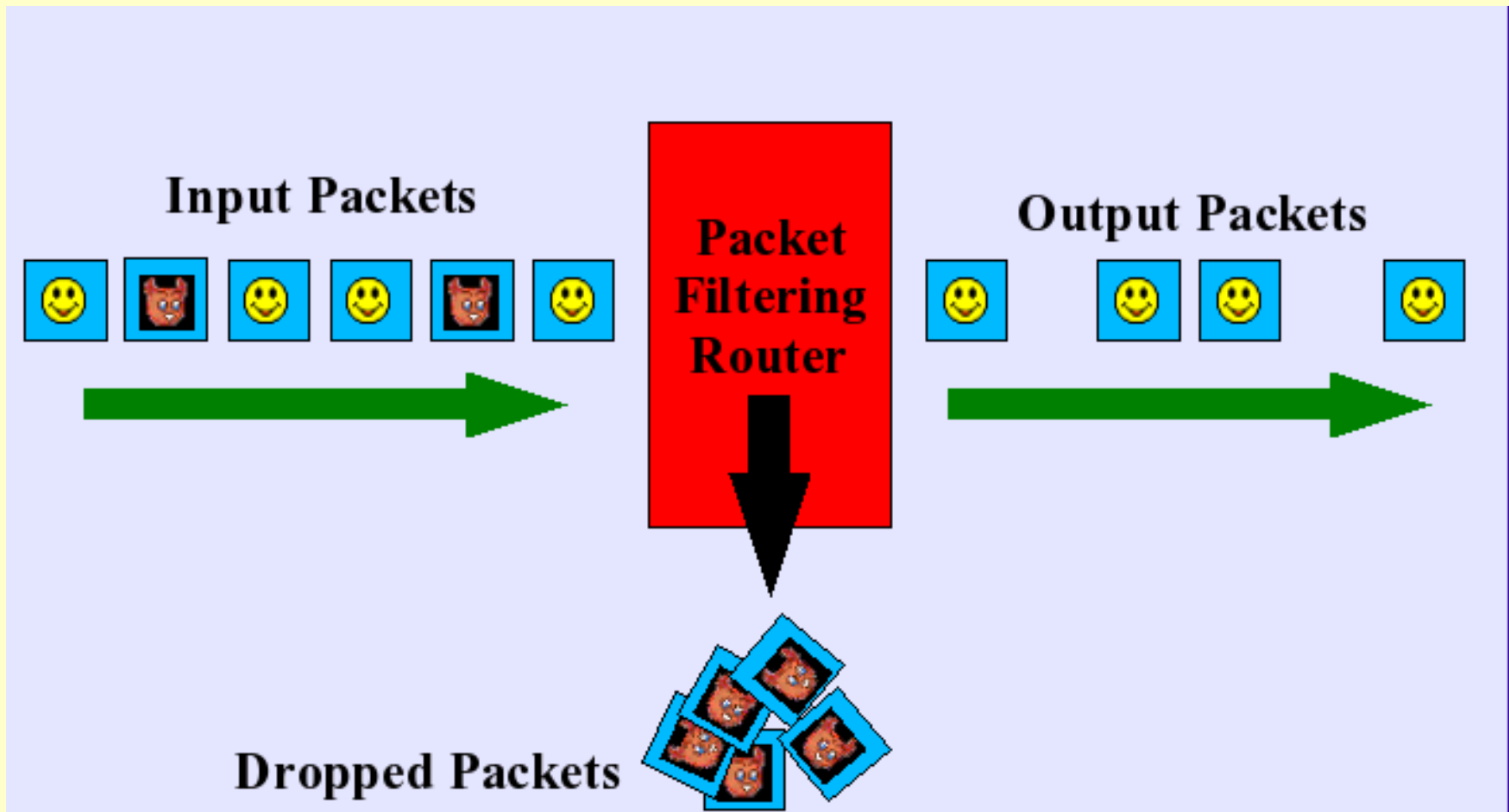
- Layer 2:
  - Source or Destination MAC addresses
- Layer 3:
  - Source or Destination addresses, ports
  - Protocol or Protocol details
    - ex: disallow IP Source Routing
    - disallow ICMP redirect packets
    - disallow common “malicious” packet signatures



# Filter on packet data

- Layer 4:
  - Service-specific (ex: by URL)
  - Structured application data (URL)
  - Unstructured application data (content)
    - Full-text scan
    - Example: email attachment types

# Packet Filtering



# Packet Filtering Rules

- Typically applied in a specific order
  - First match applies
- One filter per rule
- Default rule?
  - “Default Deny” safest
  - Warning: implied default rule: Deny or Allow?

# Example Packet Filtering

## Rules:

- Protect 128.105.0.0 network with Cisco router access control lists
- Apply rules from top to bottom:

```
deny    ip    128.105.0.0 0.0.255.255 any
permit  tcp   any 128.105.1.1 eq 25
permit  tcp   any 128.105.1.2 eq 80
permit  tcp   any 128.105.1.3 eq 22
deny    icmp  any any redirect log
permit  icmp  any 128.105.1.4 echo
deny    icmp  any any echo log
deny    ip    any any log
```



# Example Packet Filtering Rules:

- Protect 128.105.0.0 network with OpenBSD pf:

```
block in log all
```

```
block in log quick on $campus_if from  
128.105.0.0/16 to any
```

```
pass in quick on $campus_if proto tcp  
from any to 128.105.1.1/32 port = 25
```

```
...
```

```
pass in quick on $cs_if proto tcp from  
128.105.0.0/16 to any keep state
```



# Packet Filtering Advantages

- Can be placed at a few “strategic” locations
  - Internet/Internal network border
  - To isolate critical servers
- Efficient
- Simple concept



# Packet Filtering Advantages

- Widely available
  - Implemented in most routers
  - Implemented in most broadband modems and home devices
  - Implemented in some network switches
  - Firewall appliances
  - Operating systems and software
  - Specialized network interface cards with filtering capabilities



# Packet Filtering Disadvantages

- Hard to configure
  - Rules can get complex
- Hard to test and verify rules
- Incomplete implementations
- Bugs often “fail unsafe” -- allowing unintended traffic to pass



# Packet Filtering Disadvantages

- Reduces router performance
- Some policies don't map to packet filtering

# Proxy Firewalls

- Specialized application to handle specific traffic
- Protocol gateways
  - Creates new network connection, forwards data between “inside” and “outside” connection
- May apply service-specific rules and policies

# Transparent Proxies

- Not visible to sender or receiver
- Implement by intercepting and/or redirecting traffic to/from specific ports
- May complicate debugging of user problems (because proxy isn't visible to users)

# Non-Transparent Proxies

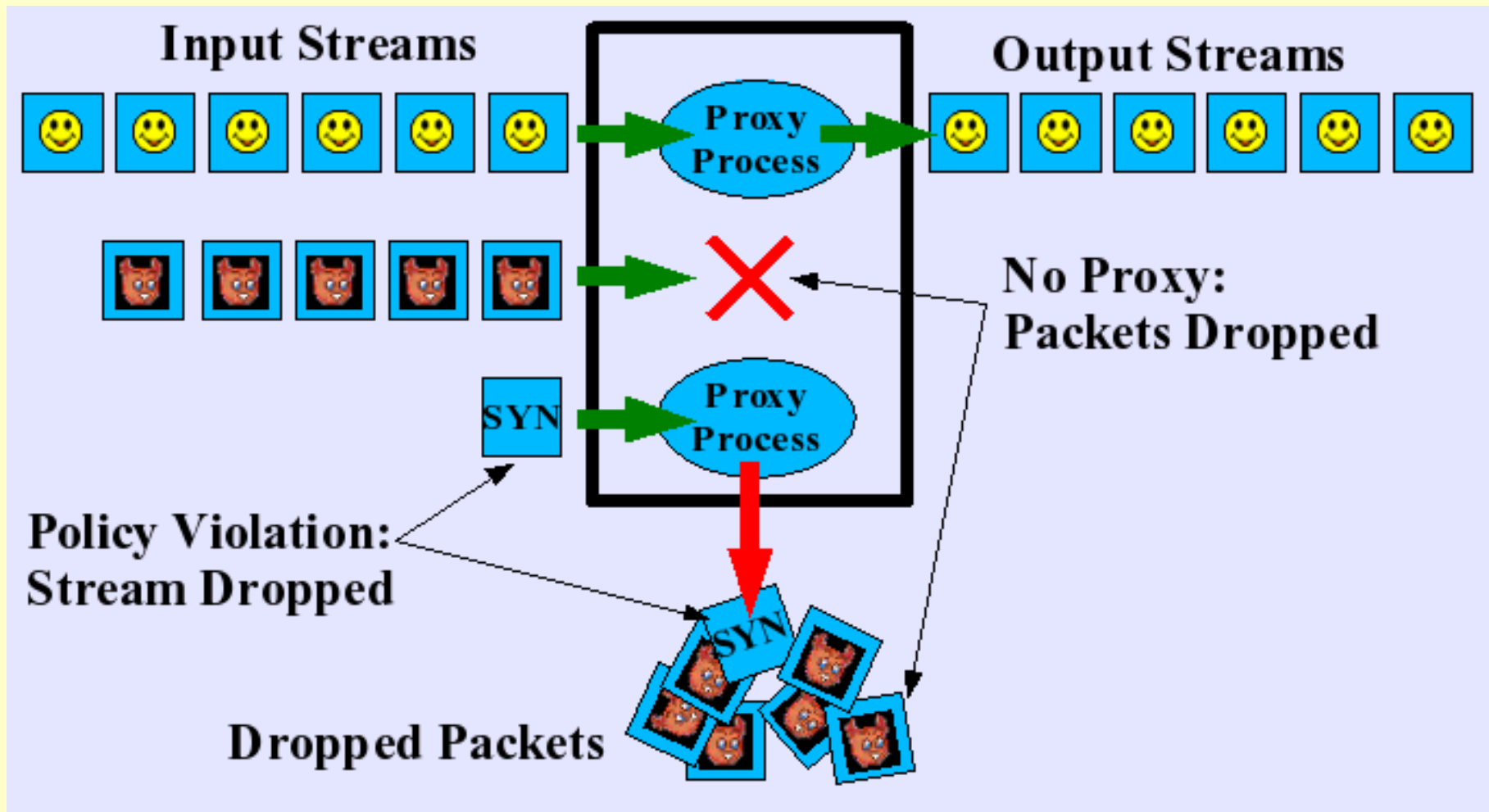
- Visible to sender and receiver
  - most likely rewrites IP headers
- Requires client reconfiguration
- May require reconfiguration at “other” end too
  - ex: change access lists to allow proxy IP, etc
- Example: Web caches/proxies



# Semi-transparent Proxies

- Not visible to client
  - no client reconfiguration necessary
- Visible to “other” end
  - all connections from proxy IP address
- Example: some web caches, load balancers
- Problems: servers that restrict access or keep state by [shared] client IP address

# Proxy Firewall



# Proxy Advantages

- Can do “intelligent” filtering
- Can perform user-level authentication
- Can use information from outside the connection or packet stream
- Can protect weak/faulty IP implementations
  - Separate network connections to source, destination

# Proxy Advantages

- Can provide application/service-specific services or actions:
  - data caching
  - data/connection logging
  - data filtering/selection or server selection based on source/destination or other status visible to proxy
  - add or apply routing/bandwidth policy



# Proxy Disadvantages

- Need to write/install proxy for each service
  - Lag time to develop proxy for new service
- May need dedicated proxy servers for each service
- Often need cooperation of clients, servers



# Dealing with Connections

- Typical scenario:
  - Restrict incoming connections to specific services and servers
    - Allow traffic to public web site
    - Allow inbound e-mail to mail gateway
  - Allow unlimited outgoing connections
    - Employees can browse the web, send e-mail, etc

# TCP Connections

- Outbound new connections often from dynamic (unpredictable) source port
  - Can't use firewall rule based on source port
- Destination may initiate another connection
  - Can't use firewall rule based on source port
- Destination may be “well-known” port
  - But not always

# TCP Connections

- Destination may move to dynamic port during connection establishment
- Issues for multi-homed servers:
  - Make sure services are listening/replying with the correct address
    - Either policy, or based on inbound packet destination address



# UDP “Connections”

- UDP is stateless
- “Connection” or “Session” implied by one or more packets from SRC to DST, one or more packets back
  - May or may not be on “well-known” port
  - May or may not be on same port as original traffic

# Handle TCP Connections Without Keeping State

- How to detect “established” TCP connections without keeping state?
  - TCP is statefull, use TCP state information
  - Established connections have ACK flag set
- “Established” keyword in many stateless firewalls allows incoming packets if ACK flag set
  - Can be exploited by faking packets with ACK flag set



# Handle UDP Connections Without Keeping State?

- Can't be done
  - UDP is stateless, not enough information in UDP packets



# Keeping State

- Stateless firewalls easy to implement
  - memory/CPU requirements are low
  - no routing impact
  - but can only act on information from the packet

# Keeping State

- Statefull/Dynamic firewalls have more information to use in decision making
  - Keeping state is more complicated, requires more CPU and memory
- Proxy Firewalls often keep state
  - But packet filtering firewalls can be statefull too

# Using State Information: TCP

- Keep Track of outbound TCP packets:
  - If match on existing session, update session data
  - If session setup packet (SYN, no ACK), create new session in state table
    - keep until session ended
  - If session shutdown packet
    - delete session from state table

# Using State Information: TCP

- Inbound TCP packets:
  - match to existing TCP session: allow packet
  - Otherwise, reject packet
- Track TCP session state, delete session from state table when finished

# Using State Information: UDP

- Keep track of outbound UDP packets:
  - If match on existing "session", update session data
  - Otherwise, create new "session" in state table
    - Keep session state for some time interval
- Inbound UDP packets:
  - Match to existing "session" -> allow packet
  - Otherwise, reject packet



# Using State Information: UDP

- Only works for typical same-port scenario
  - Reply must come from same IP as outbound traffic, go to same IP and port as outbound traffic
- More complicated session-setup protocols won't work
- No session teardown in UDP protocol, need to delete session using timeout



# Distributed Firewalls

- Two or more firewalls
  - share the load
  - redundancy in event of hardware or routing failure
- Need to keep rules synchronized
- Need to keep state synchronized

# Routing Firewalls

- Most firewall devices act as routers
- Each interface has an IP address
- Packet processing:
  - **Filters applied**
  - IP stack traversed
    - IP TTL decremented
    - Packet routed for delivery to destination



# Routing Firewalls

- Visible in network
  - TTL decremented
  - Needs to be in routing table of immediate neighbors
  - Shows in traceroute (TTL decremented)

# Bridging Firewalls

- “Bump in the road”
- Interfaces do not have IP addresses
- Packet processing:
  - **Filters applied**
  - No IP stack in firewall path
    - IP TTL NOT decremented
  - Packet forwarded (unchanged) towards destination



# Bridging Firewalls

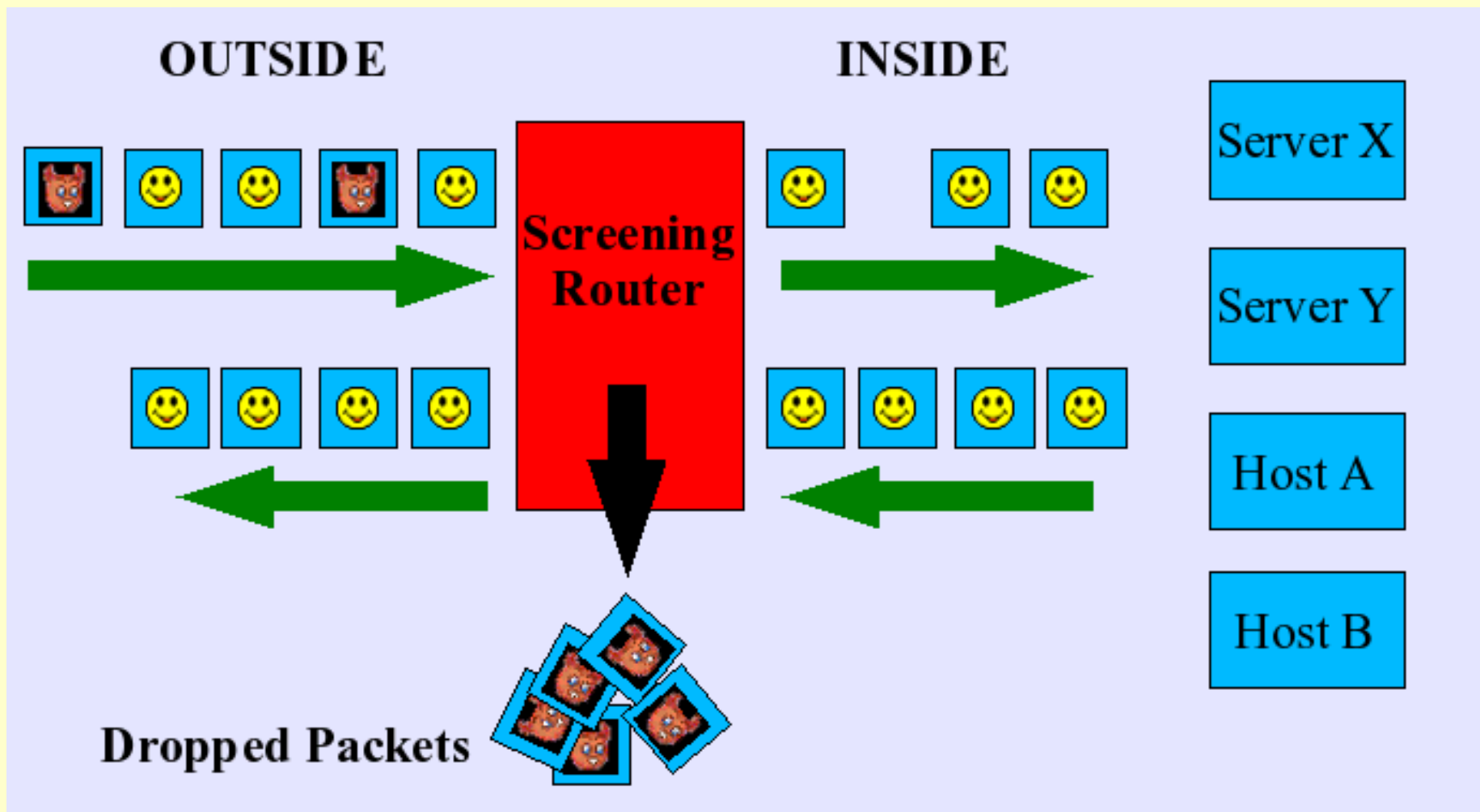
- Not visible in network
- No changes in neighbor configuration
- Not visible in traceroute
- Debugging more difficult



# Firewall Deployment Scenarios

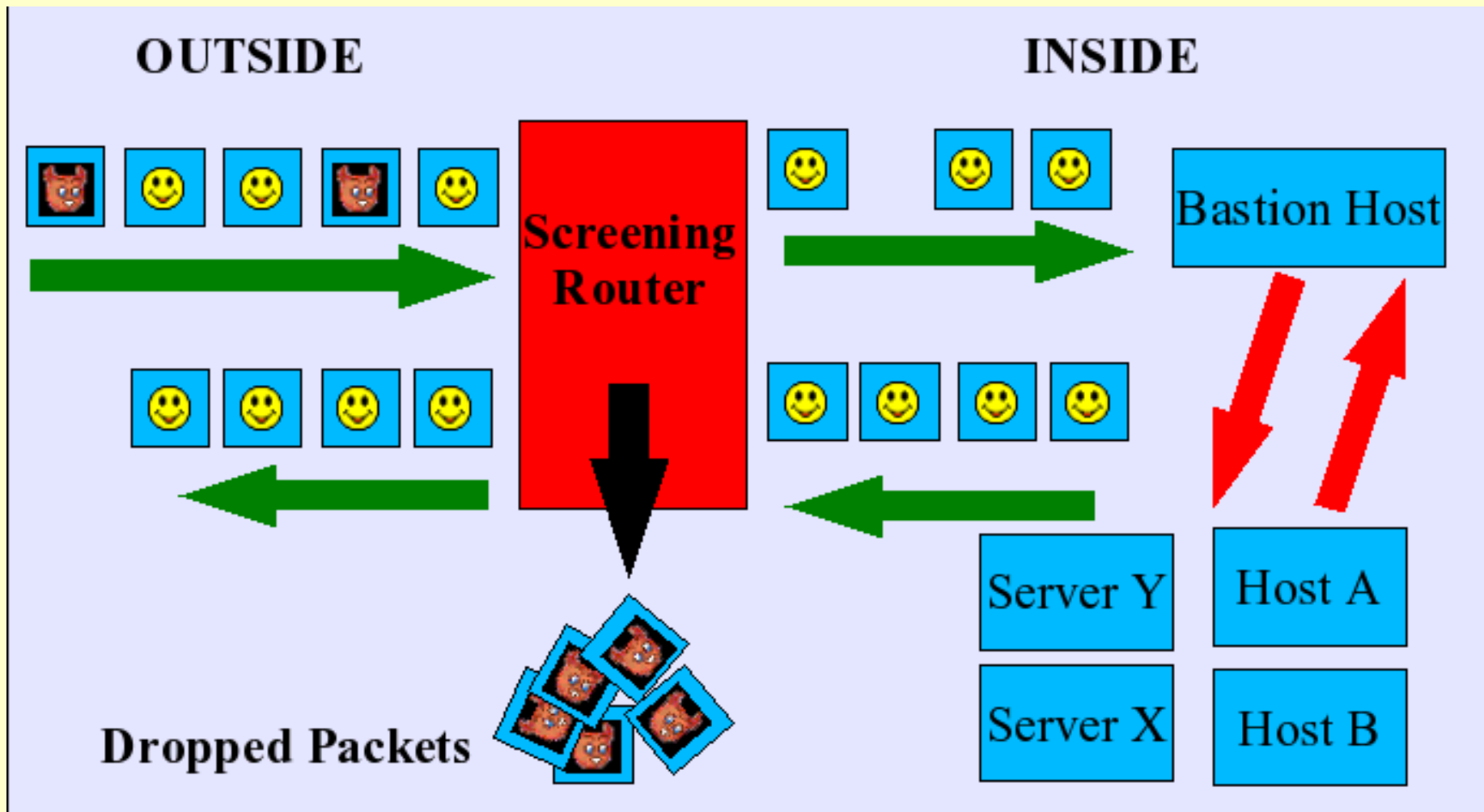
- Screening router
- Screening router with “bastion host”
- Screened subnet (“DMZ”)
- Internal firewalls
- Host-based firewalls
- Multiple Variations

# Screening Router

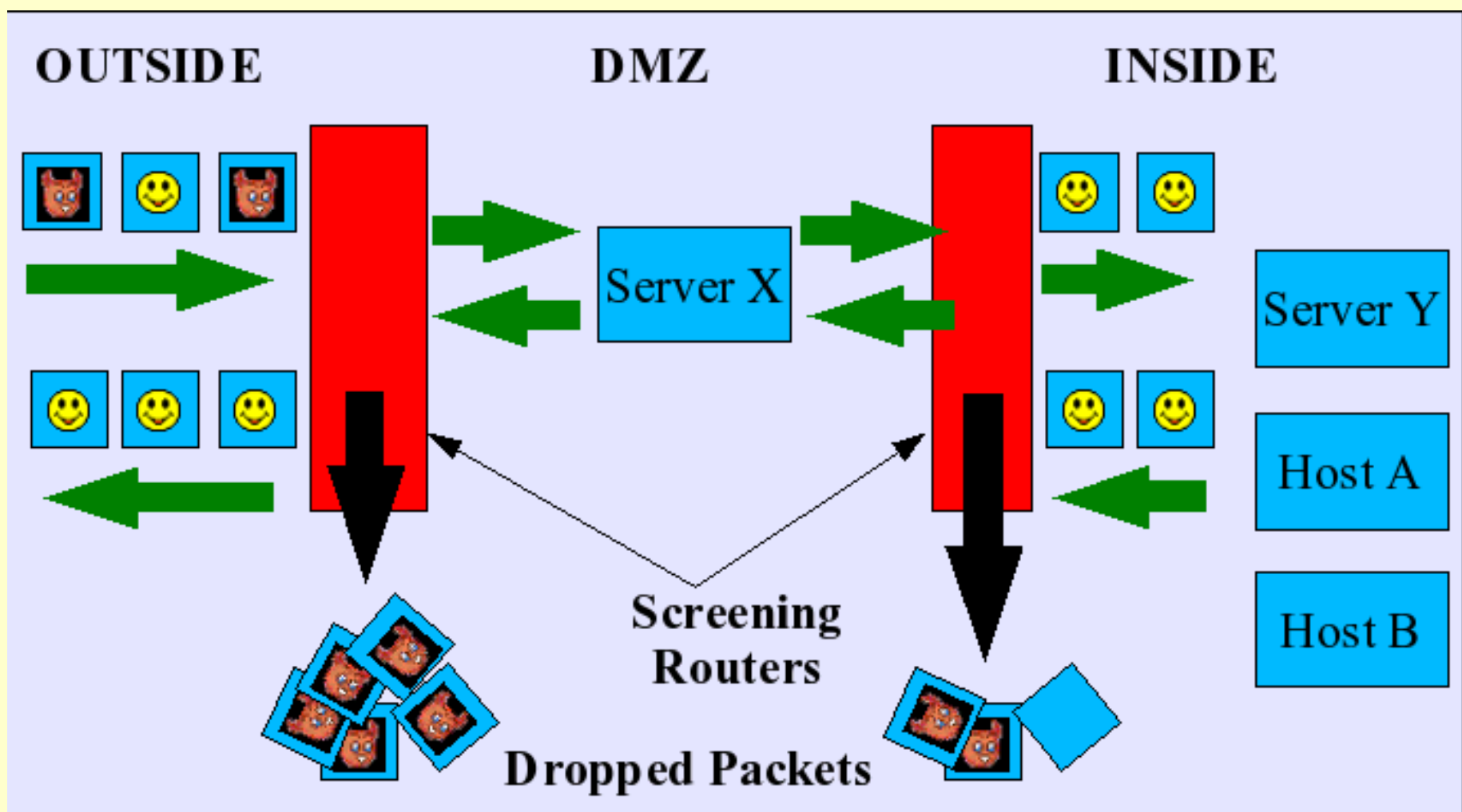




# Screening Router with Bastion Host



# Screened Subnet ("DMZ")





# Internal Firewalls

- More common in larger organizations
- Not safe to assume that all “bad guys” are outside
- Prevent accidents, isolate damage
- Apply appropriate security policies to selected servers/areas of operation
  - Smaller, specialized perimeters



# Internal Firewalls

- Separate internal operations should be isolated on the network
  - Example: Purchasing and Accounts Payable
  - Example: isolate credit card processing operations
  - Different parts of the organization have relationships with different outside groups
  - Outside groups may be competitors, require isolation from each other



# Host-Based Firewalls

- Typically implemented as part of the OS
  - Not visible in the network (like an internal bridge)
- Typically provide packet-filtering
  - May provide logging and packet capture facilities



# Host-Based Firewalls: Pros

- Provide an additional line of defense
  - Narrow the perimeter from network segment to single host
  - Diversity of firewall implementation
    - May cover errors/gaps in firewall implementation
  - Diversity of security policy expression
    - May cover errors/gaps in expressing security policy by restating policy using different firewall configuration language

# Host-Based Firewalls: Cons

- If the host is compromised, local firewall is compromised
  - Users have access to the host
  - Other services on the host: possible access points
- Relies on the same code base as the rest of the OS: OS/library bugs may impact firewall



# Host-Based Firewalls: Cons

- Complexity of managing many per-host configurations
  - Tens, Hundreds, perhaps Thousands
  - Can be automated, depending on the site





# Application Firewalls

- Network access control rules for specific applications/services
  - Like other firewalls, rules can be based on packet data or application-specific content
  - In addition to any network or host-based firewalls
  - May allow application-specific customizations
- e.g.: tcpwrappers for linux/unix services



# Related Technologies

- Network Address Translation
- Virtual Private Networks
- Active Defenses

# Network Address Translation

- Specialized proxy
  - Rewrites IP addresses, ports
  - Map “private” IP addresses to “public” addresses
    - Conserve IP address space
    - RFC 1918
  - Typical configuration for home broadband
    - Common for public wifi
  - Also used for virtual servers, load balancing



# Network Address Translation

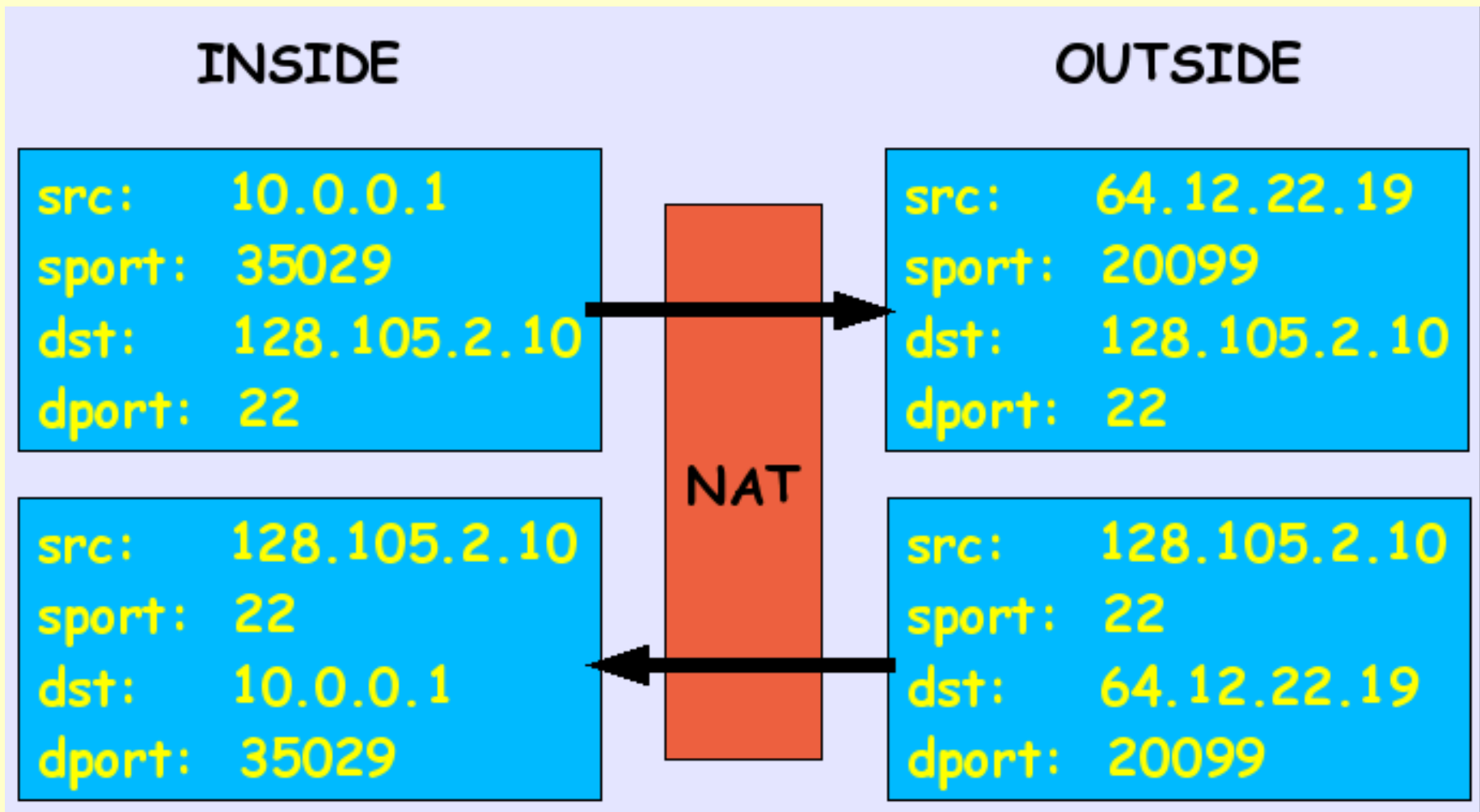
- Protects unmapped “inside” addresses
  - not visible at all to “outside” addresses



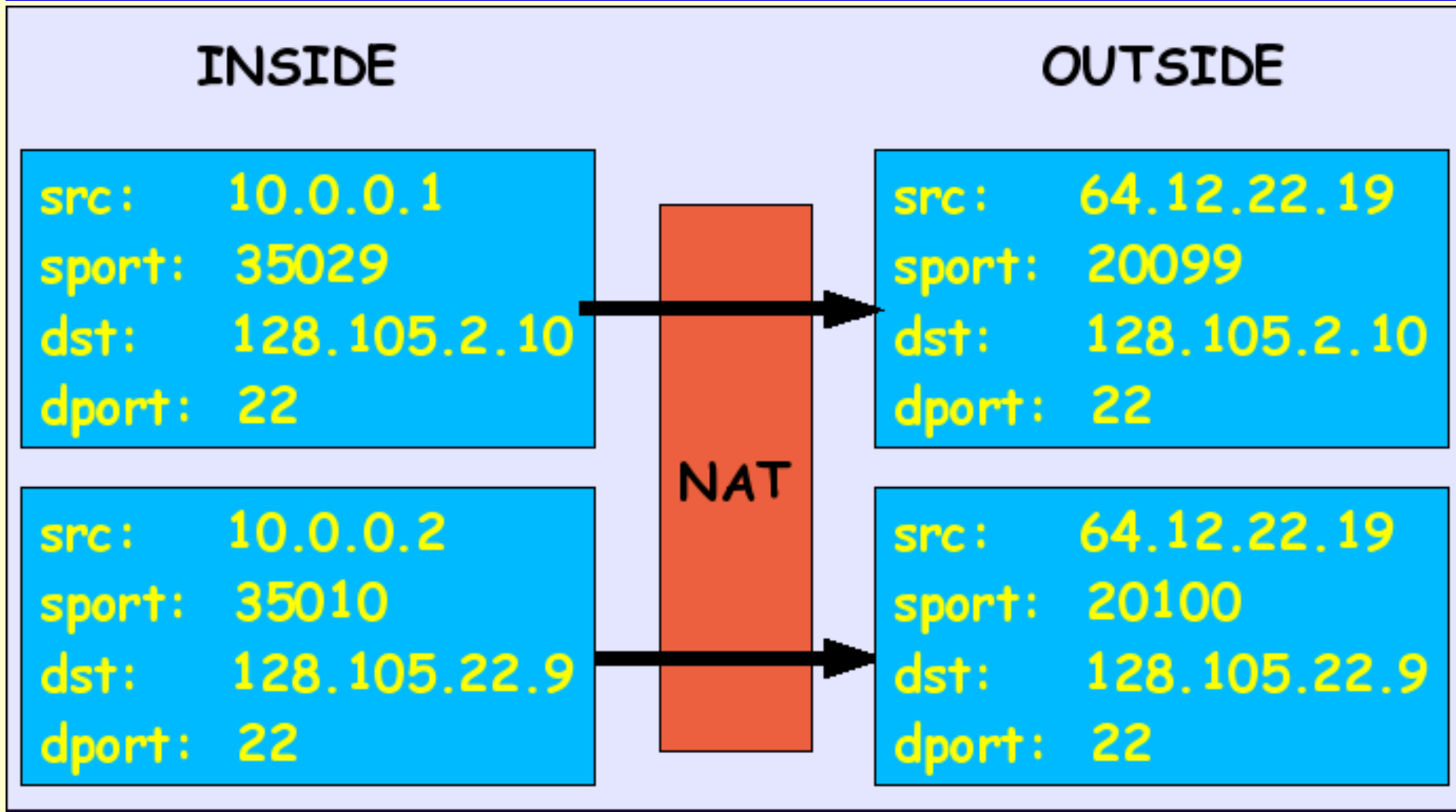
# Network Address Translation

- Implemented in most home broadband routers/modems
  - 1 IP address from broadband network
  - multiple computers and IP addresses “inside” home network
  - limited capability to specify “inside” addresses/ports to expose to “outside”

# Network Address Translation



# Network Address Translation



# Virtual Private Networks (VPNs)

- Tunnel traffic from host/network A to host/network B
  - Encapsulate in something else (IP, SSH, IP SEC, etc)
  - Usually includes encryption, authentication
- Block all external traffic except to “public” services
- Allow only VPN traffic to internal services



# Virtual Private Networks (VPNs)

- Danger: VPN traffic usually bypasses firewall...
- VPN can allow "outside" traffic to bypass firewall
  - Other systems at home may incorrectly route via VPN
- Can lower the "inside" security standard
  - Other home systems may not be patched...

# Virtual Private Networks (VPNs)

- Problem: Breaks the “inside/outside” simple security model: harder to explain to people
- Breaks assumptions that may or may not have been valid or correct in the first place

# Virtual Private Networks (VPNs)

- Example problem: UW Libraries
  - Outmoded license restrictions imposed by content vendors
  - Faculty, staff, students should be given access, on-campus or off-campus (via the VPN)
  - Other VPN users (visitors) *on campus* should get access
  - Other VPN users (visitors) *off campus* should **not** get access

# Virtual Private Networks (VPNs)

- Can't distinguish physical location at the network level
  - Mostly not a technology problem
  - Should not have been using IP address for authorization
    - But that is what the vendors use and understand

# Active Defenses

- Intrusion Detection/Prevention Systems can respond to observed traffic:
  - Generate spontaneous FIN to kill “bad” TCP session
  - Generate feedback to the firewall to block traffic from “bad” sites
- Requires well-tuned IDS and active monitoring by staff to avoid or mitigate false positives



# Real World Experience

- CS Firewall
- Home Firewalls

# CS Border Firewall

- “Trip Curb”
  - You can stub your toe if you kick it
  - Rules getting more complex... the curb is taller and more solid now
- Screening/Packet Filtering firewall
  - statefull
  - OpenBSD bridging firewall

# CS Border Firewall: Input Rules

- Default “allow”
  - Block known problem ports
  - Block unneeded services with potential problems
    - NFS, RPC, NETBIOS ...
- Block forged/malformed packets
  - Inbound with our SRC address
  - Inbound with “unrouteable” SRC addresses



# CS Border Firewall: Input Rules

- Enforce some policies
  - SMTP only to mail gateways (virus scanning)
  - WWW only to known web servers
- Allow inbound packets for established connections/sessions (statefull)
- Block all traffic to special networks



# CS Border Firewall: Output Rules

- Block forged/malformed packets
  - Outbound without our SRC address
- Block all traffic from special networks



# CS Border Firewall: Next Steps

- Switch to “default deny”
- Better analysis tools

# Other CS Firewalls

- Host-based firewalls
- Un-patched/Experimental network
  - Can only reach other CS networks
  - Can not send/receive email (even inside CS)
- Crash-and-Burn network
  - Can only reach other CS networks
  - Some services restricted

# Other CS Firewalls

- Wireless/Laptop network
  - Can only do DNS until authenticated
  - After authentication, allow traffic that was initiated by wifi clients
- Install network
  - Used by CSL for installing OS on new computers
  - Isolated from internet to prevent attacks before OS installation/patching complete

# Home Firewalls

- Strongly advised for any high-speed internet connection (DSL, Cable Modem, etc)
- Simple NAT/Packet Filtering appliances
  - \$35-150
  - Or included in broadband modem
- Build your own: Linux, OpenBSD, ...

# Firewall Strengths

- Relatively inexpensive to deploy
- Easy to understand the idea, maps to simple business requirements
  - “Inside” and “Outside”
- Identifiable security point
- Limits exposure
- Weaknesses are known
- Can find good firewall implementations



# Firewall Weaknesses

- Interferes with the way the Internet protocols are designed to work
  - Can interrupt service
  - Delay in introducing new services
- Ignores internal threats, other entry points





# Firewall Weaknesses

- Complicated to manage in complicated environments
- Any kind of tunneling (VPN, SSH, etc) has to bypass firewall
- Potential false sense of security



# Summary

- Firewalls can be an effective tool
  - Need to understand limitations
  - Need to review configuration, policy on regular basis
- Internal firewalls gaining hold
- Use with other security tools, with a more complete security policy



# Questions?