Firewalls

What is a Firewall?

- A **choke point** of control and monitoring
- Interconnects networks with differing trust
- Imposes restrictions on network services
  - only authorized traffic is allowed
- Auditing and controlling access
  - can implement alarms for abnormal behavior
- Itself immune to penetration
- Provides **perimeter defence**

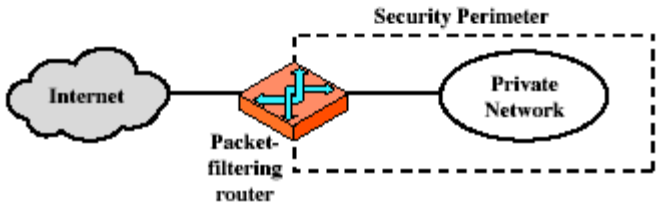
Classification of Firewall

Characterized by protocol level it controls in

- Packet filtering
- Circuit gateways
- Application gateways

- Combination of above is dynamic packet filter

Firewalls – Packet Filters



(a) Packet-filtering router

Firewalls – Packet Filters

- Simplest of components
- Uses transport-layer information only
  - IP Source Address, Destination Address
  - Protocol/Next Header (TCP, UDP, ICMP, etc)
  - TCP or UDP source & destination ports
  - TCP Flags (SYN, ACK, FIN, RST, PSH, etc)
  - ICMP message type
- Examples
  - DNS uses port 53
    - No incoming port 53 packets except known trusted servers

Usage of Packet Filters

- Filtering with incoming or outgoing interfaces
  - E.g., Ingress filtering of spoofed IP addresses
  - Egress filtering
- Permits or denies certain services
  - Requires intimate knowledge of TCP and UDP port utilization on a number of operating systems

How to Configure a Packet Filter

- Start with a security policy
- Specify allowable packets in terms of logical expressions on packet fields
- Rewrite expressions in syntax supported by your vendor
- General rules - least privilege
  - All that is not expressly permitted is prohibited
  - If you do not need it, eliminate it

Every ruleset is followed by an implicit rule reading like this.

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

Example 1:

Suppose we want to allow inbound mail (SMTP, port 25) but only to our gateway machine. Also suppose that mail from some particular site SPIGOT is to be blocked.

Solution 1:

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

Example 2:

Now suppose that we want to implement the policy "any inside host can send mail to the outside".

Solution 2:

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

This solution allows calls to come from any port on an inside machine, and will direect them to port 25 on the outside. Simple enough…

So why is it wrong?

- Our defined restriction is based solely on the outside host's port number, which we have no way of controlling.
- Now an enemy can access any internal machines and port by originating his call from port 25 on the outside machine.

Now for a better solution…

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	25		*our packets to their SMTP port*
allow	*	25	*	*	`ACK`	*their replies*

- The ACK signifies that the packet is part of an ongoing conversation
- Packets without the ACK are connection establishment messages, which we are only permitting from internal hosts

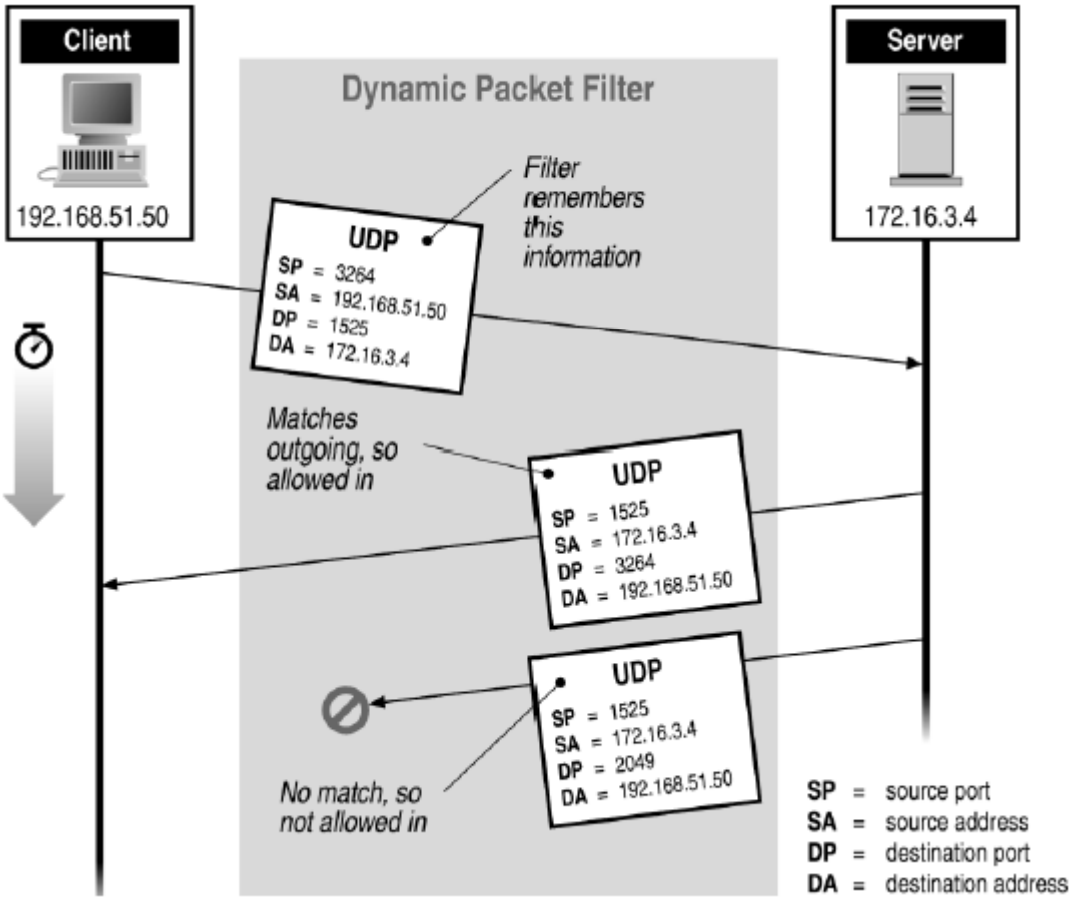
Security & Performance of Packet Filters

- IP address spoofing
  - Fake source address to be trusted
  - Add filters on router to block
- Degradation depends on number of rules applied at any point
- Order rules so that most common traffic is dealt with first
- Correctness is more important than speed

Port Numbering

- TCP connection
  - Server port is number less than 1024
  - Client port is number between 1024 and 16383
- Permanent assignment
  - Ports <1024 assigned permanently
    - 20,21 for FTP 23 for Telnet
    - 25 for server SMTP 80 for HTTP
- Variable use
  - Ports >1024 must be available for client to make any connection
  - This presents a limitation for stateless packet filtering
    - If client wants to use port 2048, firewall must allow *incoming* traffic on this port
  - Better: stateful filtering knows outgoing requests

Stateful Filtering



Firewall Outlines

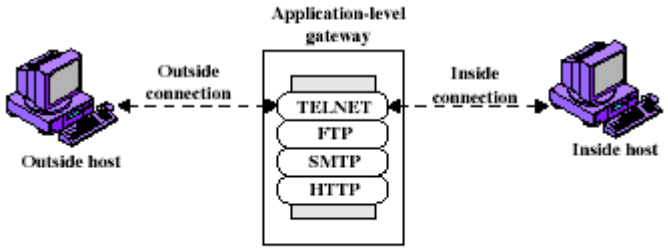
- Packet filtering
- Application gateways
- Circuit gateways

- Combination of above is dynamic packet filter

Firewall Gateways

- Firewall runs set of proxy programs
  - Proxies filter incoming, outgoing packets
  - All incoming traffic directed to firewall
  - All outgoing traffic appears to come from firewall
- Policy embedded in proxy programs
- Two kinds of proxies
  - Application-level gateways/proxies
    - Tailored to http, ftp, smtp, etc.
  - Circuit-level gateways/proxies
    - Working on TCP level

Firewalls - Application Level Gateway (or Proxy)

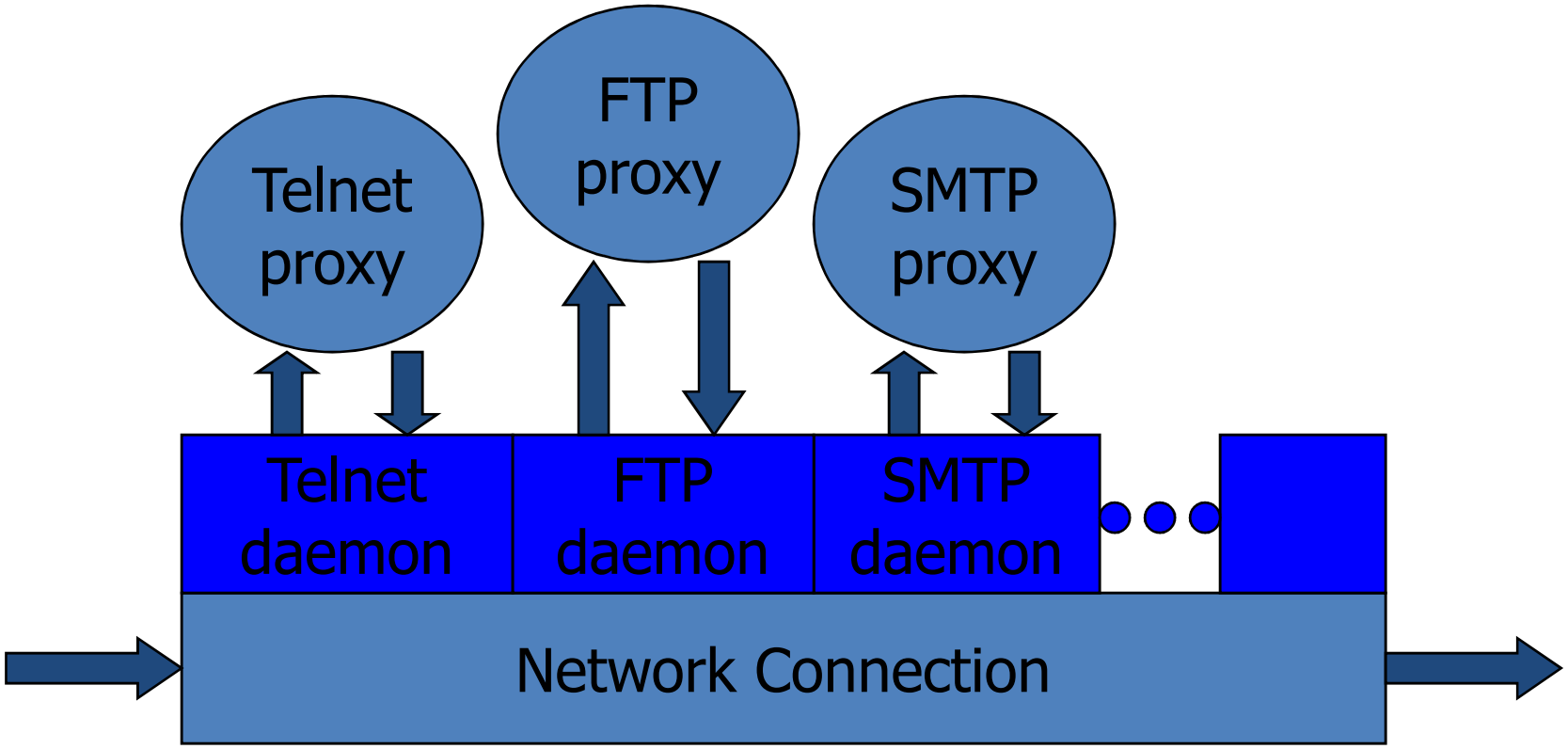


(b) Application-level gateway

Application-Level Filtering

- Has full access to protocol
  - user requests service from proxy
  - proxy validates request as legal
  - then actions request and returns result to user
- Need separate proxies for each service
  - E.g., SMTP (E-Mail)
  - NNTP (Net news)
  - DNS (Domain Name System)
  - NTP (Network Time Protocol)
  - custom services generally not supported

App-level Firewall Architecture



Daemon spawns proxy when communication detected …

Enforce policy for specific protocols

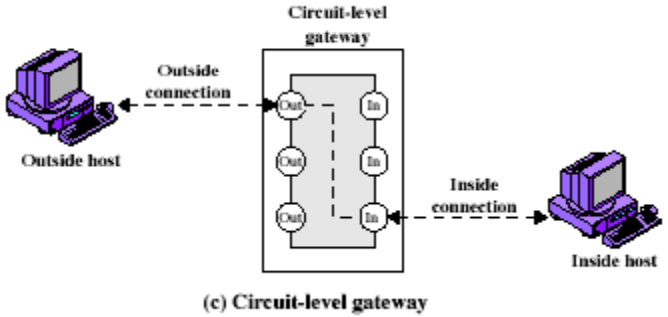
- E.g., Virus scanning for SMTP
  - Need to understand MIME, encoding, Zip archives

Firewall Outlines

- Packet filtering
- Application gateways
- Circuit gateways

- Combination of above is dynamic packet filter

Firewalls - Circuit Level Gateway



(c) Circuit-level gateway

Firewalls - Circuit Level Gateway

- Relays two TCP connections
- Imposes security by limiting which such connections are allowed
- Once created usually relays traffic without examining contents
- Typically used when trust internal users by allowing general outbound connections
- SOCKS commonly used for this

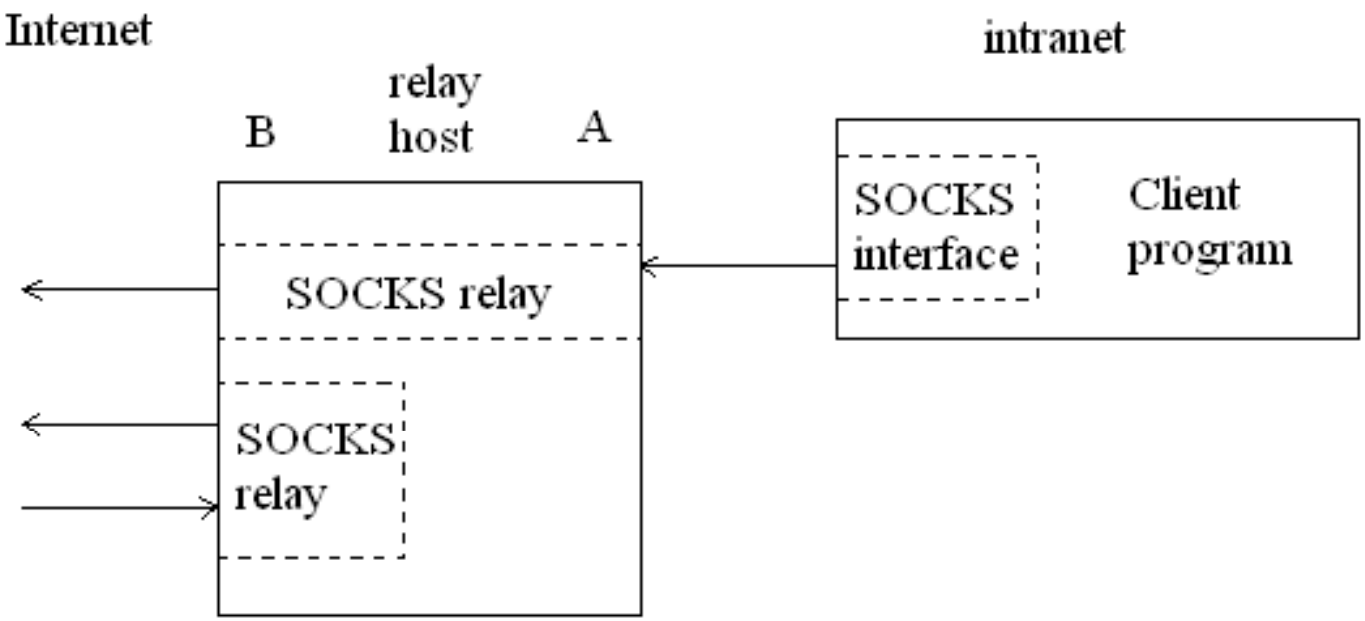
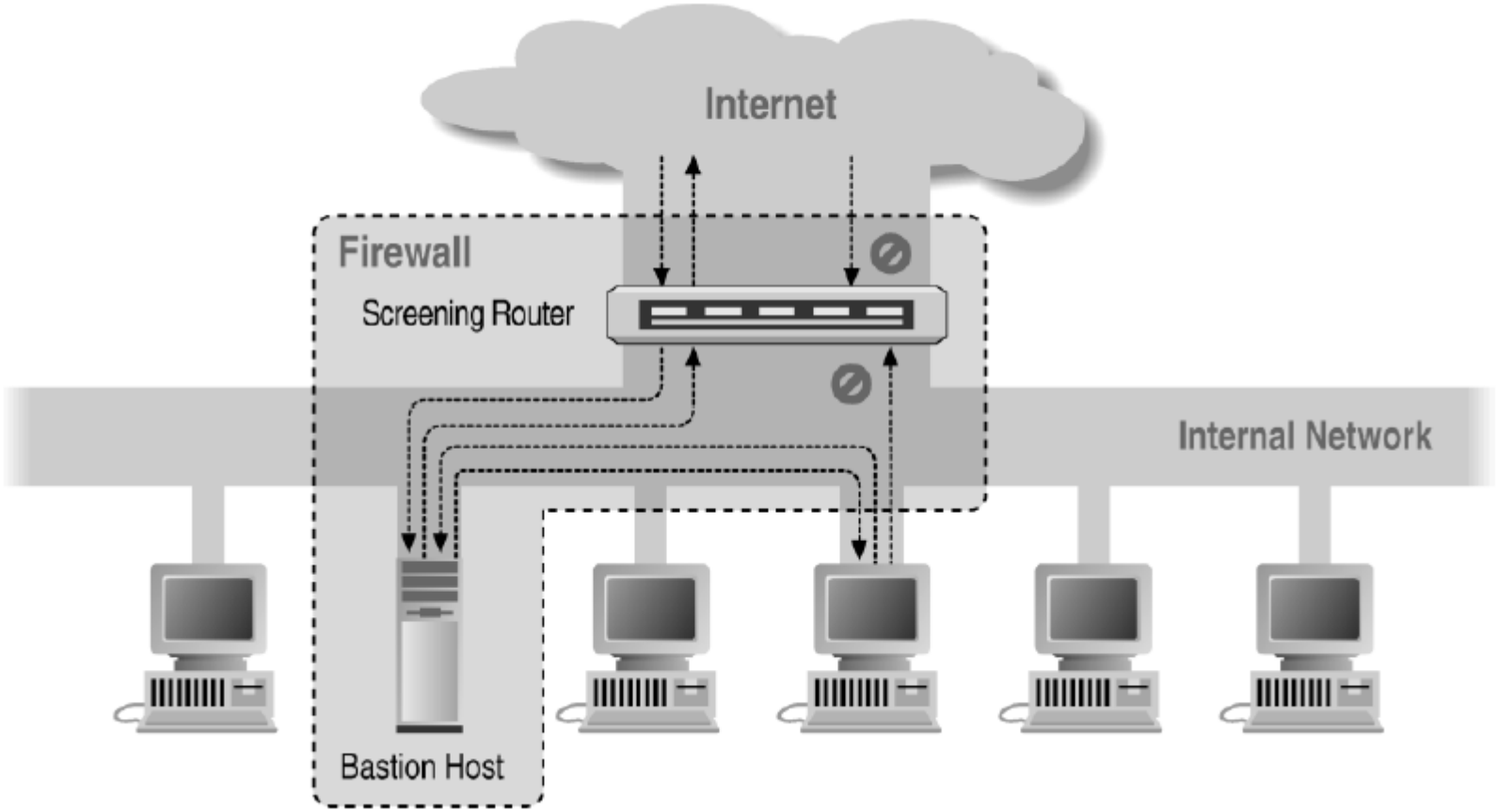


Figure 9.7: A typical SOCKS connection through interface A, and rogue connection through the external interface, B.

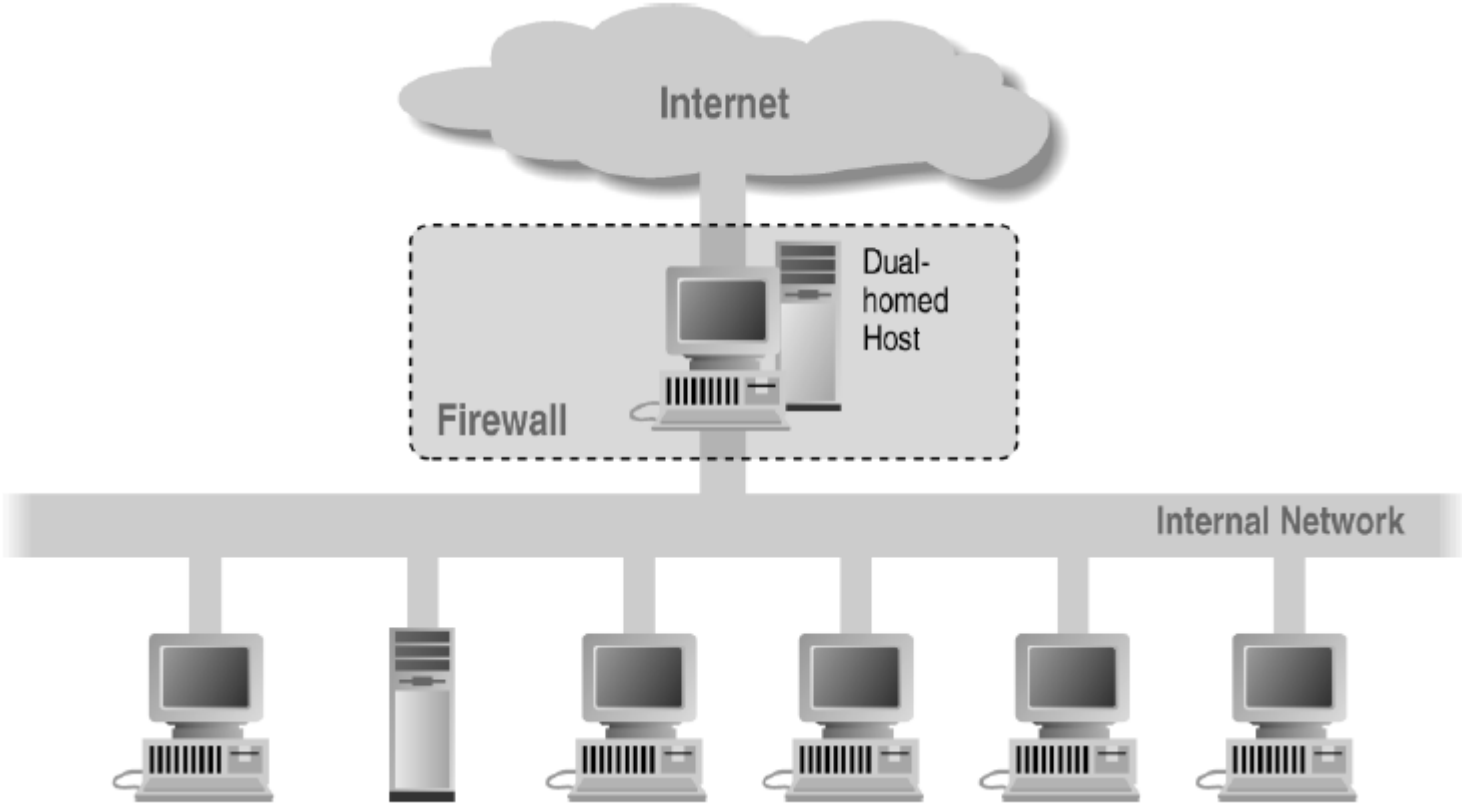
Bastion Host

- Highly secure host system
- Potentially exposed to "hostile" elements
- Hence is secured to withstand this
  - Disable all non-required services; keep it simple
- Trusted to enforce trusted separation between network connections
- Runs circuit / application level gateways
  - Install/modify services you want
- Or provides externally accessible services

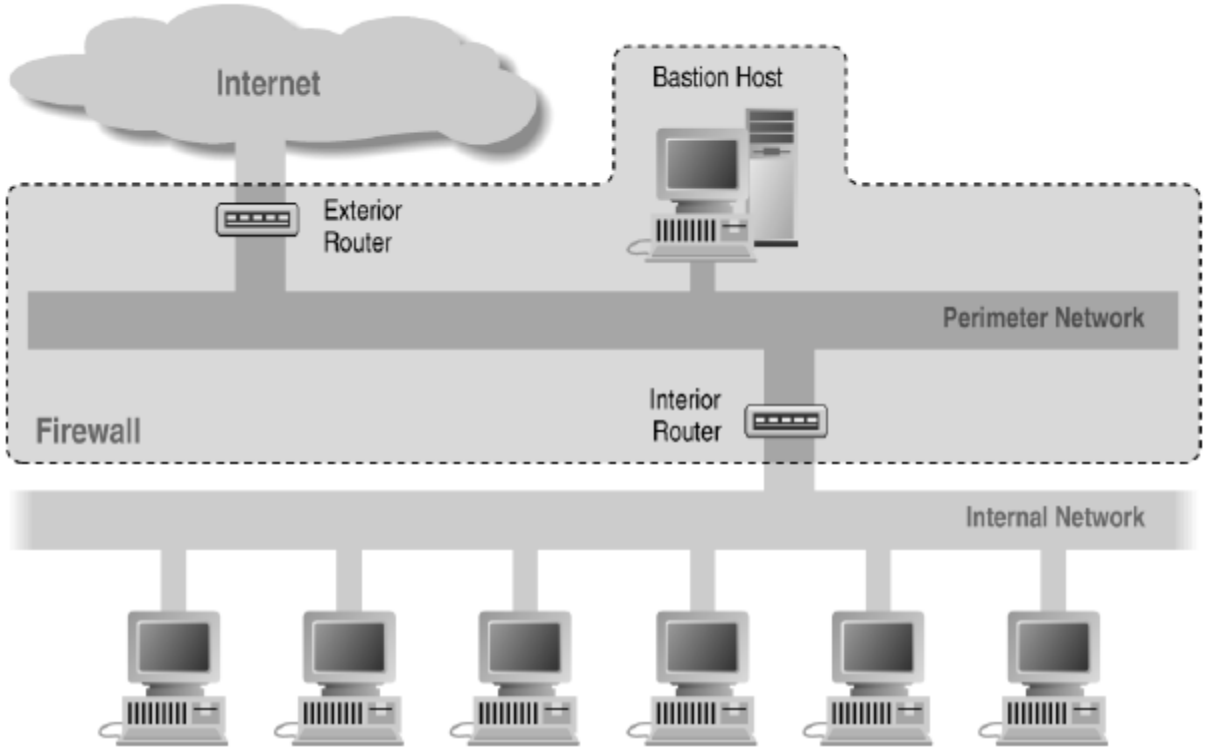
Screened Host Architecture



Dual Homed Host Architecture



Screened Subnet Using Two Routers



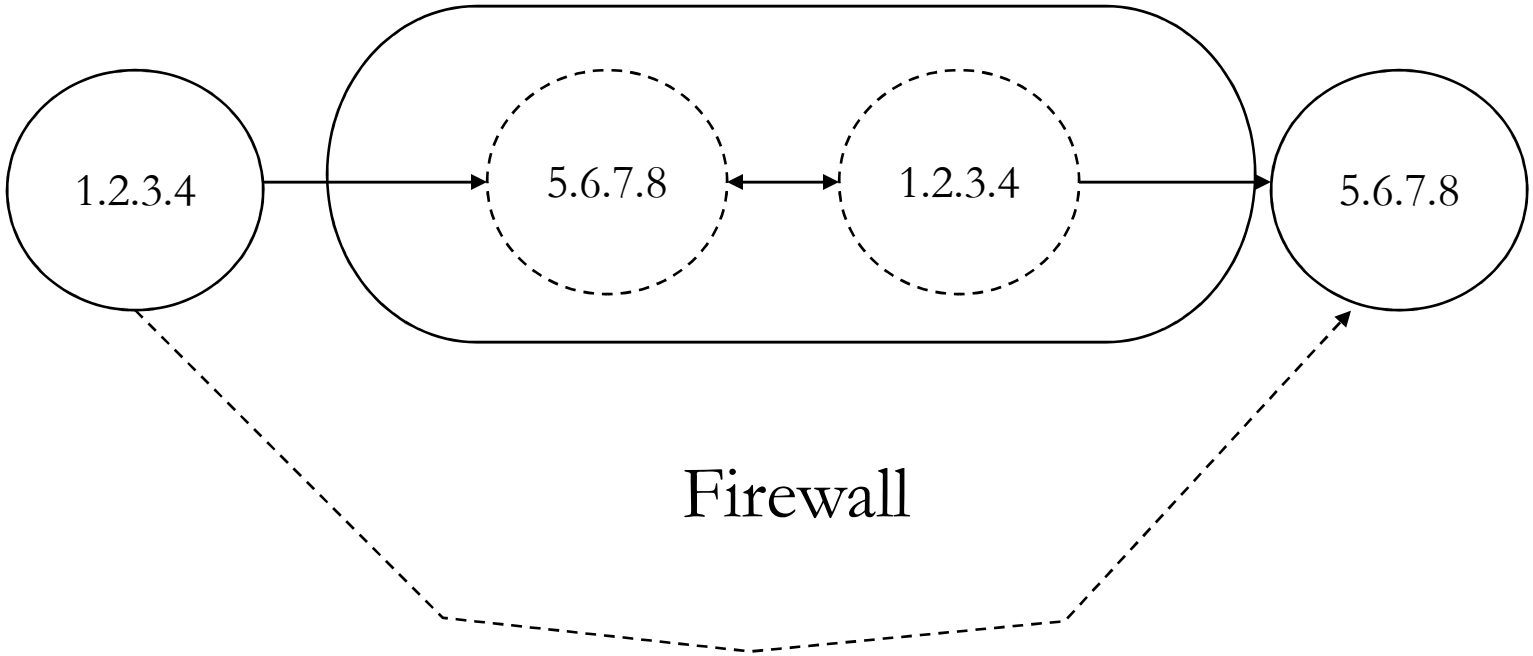
Firewall Outlines

- Packet filtering
- Application gateways
- Circuit gateways

- Combination of above is dynamic packet filter

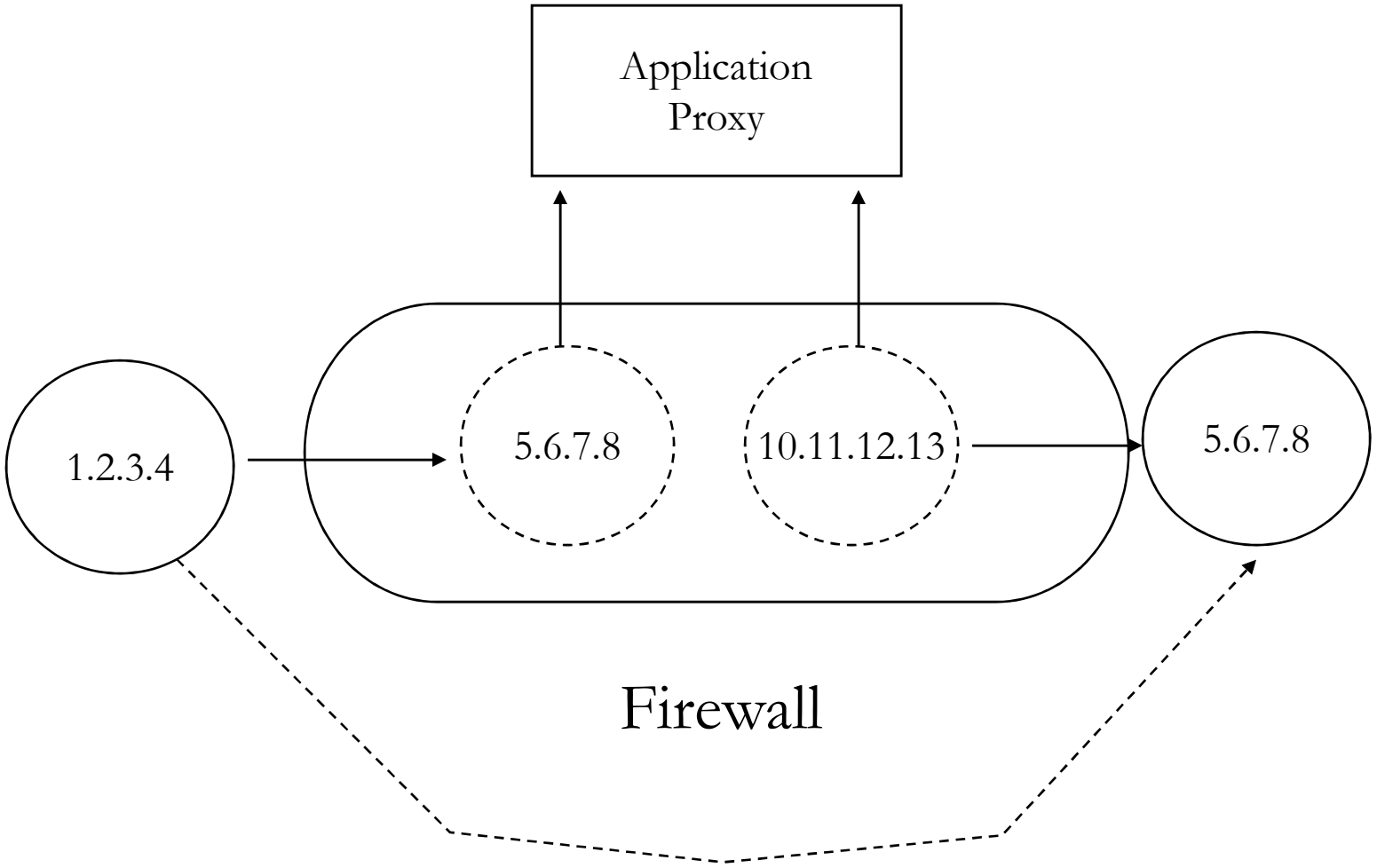
Dynamic Packet Filters

- Most common
- Provide good administrators protection and full transparency
- Network given full control over traffic
- Captures semantics of a connection



Intended connection from 1.2.3.4 to 5.6.7.8

Redialing on a dynamic packet filter. The dashed arrow shows the intended connection; the solid arrows show the actual connections, to and from the relay in the firewall box. The Firewall impersonates each endpoint to the other.



A dynamic packet filter with an application proxy. Note the change in source address

Dynamic Packet Filter Implementation

- Dynamically update packet filter's ruleset
  - Changes may not be benign due to ordering
- Redialing method offers greater assurance of security
  - No special-case code necessary
  - FTP handled with user-level daemon
  - UDP handled just as TCP except for tear down
  - ICMP handled with pseudoconnections and synthesized packets

Per-Interface Tables Consulted by Dynamic Packet Filter

- Active Connection Table
  - Socket structure decides whether data is copied to outside socket or sent to application proxy
- Ordinary Filter Table
  - Specifies which packets may pass in stateless manner
- Dynamic Table
  - Forces creation of local socket structures

Asymmetric Routes

- Both sides of the firewall know nothing of one another's topology
- Solutions:
  - Maintain full knowledge of the topology
    - Not feasible, too much state to keep
  - Multiple firewalls share state information
    - Volume of messages may be prohibitive, code complexity

Are Dynamic Packet Filters Safe?

- Comparable to that of circuit gateways, as long as the implementation strategy is simple
- If administrative interfaces use physical network ports as the highest-level construct
  - Legal connections are generally defined in terms of the physical topology
- Not if evildoers exist on the inside
  - Circuit or application gateways demand user authentication for outbound traffic and are therefore more resistant to this threat

Distributed Firewalls

- A central management node sets the security policy enforced by individual hosts
- Combination of high-level policy specification with file distribution mechanism
- Advantages:
  - Lack of central point of failure
  - Ability to protect machines outside topologically isolated space
  - Great for laptops
- Disadvantage:
  - Harder to allow in certain services, whereas it's easy to block

Distributed Firewalls Drawback

- Allowing in certain services works if and only if you're sure the address can't be spoofed
  - Requires anti-spoofing protection
  - Must maintain ability to roam safely
- Solution: IPsec
  - A machine is trusted if and only if it can perform proper cryptographic authentication

Where to Filter?

- Balance between risk and costs
- Always a higher layer that is hard to filter
  - Humans

Firewalls Aren't Perfect?

- Useless against attacks from the inside
  - Evildoer exists on inside
  - Malicious code is executed on an internal machine
- Organizations with greater insider threat
  - Banks
  - Military
- Protection must exist at each layer
  - Assess risks of threats at every layer
- Rely on transitive trust

Backup Slides

Network Topology

Filter Rule: Open access to Net 2 means source address from Net 3

- Why not spoof address from Net 3?

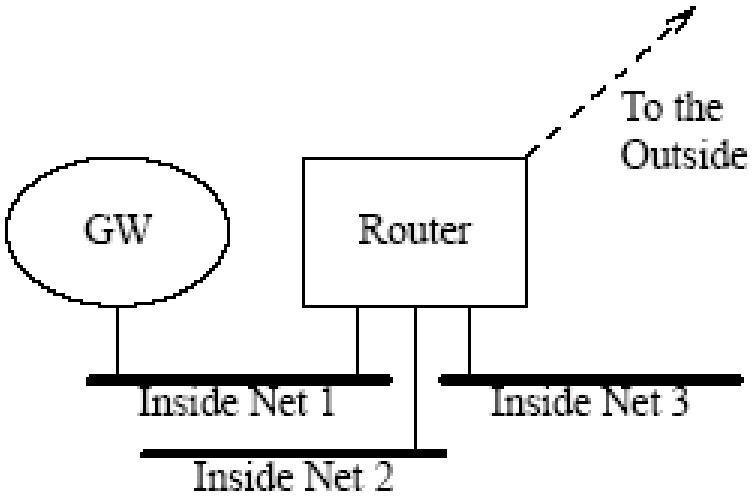


Figure 9.2: A firewall router with multiple internal networks.

Address-Spoofing

- Detection is virtually impossible unless source-address filtering and logging are done
- One should not trust hosts outside of one's administrative control

External Interface Ruleset

Allow outgoing calls, permit incoming calls only for mail and only to gateway GW

action	src	port	dest	port	flags	comment
block	{NET 1}	*	*	*		*block forgeries*
block	{NET 2}	*	*	*		
block	{NET 3}	*	*	*		
allow	*	*	GW	25		*legal calls to us*
allow	*	*	{NET 2}	*	ACK	*replies to our calls*
allow	*	*	{NET 3}	*	ACK	

Note: Specify GW as destination host instead of Net 1 to prevent open access to Net 1

Net 1 Router Interface Ruleset

- Gateway machine speaks directly only to other machines running trusted mail server software
- Relay machines used to call out to GW to pick up waiting mail

action	src	port	dest	port	flags	comment
allow	GW	*	{*partners*}	25		*mail relay*
allow	GW	*	{NET 2}	*	ACK	*replies to inside calls*
allow	GW	*	{NET 3}	*	ACK	
block	GW	*	{NET 2}	*		*stop other calls from* GW
block	GW	*	{NET 3}	*		
allow	GW	*	*	*		*let* GW *call the world*

Note: Spoofing is avoided with the specification of GW

How Many Routers Do We Need?

- If routers only support outgoing filtering, we need two:
  - One to use ruleset that protects against compromised gateways
  - One to use ruleset that guards against address forgery and restricts access to gateway machine
- An input filter on one port is exactly equivalent to an output filter on the other port
- If you trust the network provider, you can go without input filters
  - Filtering can be done on the output side of the router

Routing Filters

- All nodes are somehow reachable from the Internet
- Routers need to be able to control what routes they advertise over various interfaces
- Clients who employ IP source routing make it possible to reach ‘unreachable’ hosts
  - Enables address-spoofing
  - Block source routing at borders, not at backbone

Routing Filters (cont)

- Packet filters obviate the need for route filters
- Route filtering becomes difficult or impossible in the presence of complex technologies
- Route squatting – using unofficial IP addresses inside firewalls that belong to someone else
- Difficult to choose non-addressed address space