Firewalls

Controlling Networks

- motivation
  - harden a network against external attack
  - the more public facing network services you run the greater the risk
  - MINIMIZE ATTACK SURFACE
- one approach: disable services you don't need
  - you may be running some you don't realize
  - sometimes you need to allow trusted remote users in
  - hard to scale
    - you have hundreds or thousands of systems and services
    - different OSs, hardware, etc.

Reducing Complexity

- reduce risk by blocking **outsiders** from accessing network
- put a **firewall** that monitors and controls all traffic to and from the outside
  - single point that can "disable services" for thousands of hosts
  - e.g., if a security threat in a popular software is found

Firewall Security Policy

- effectiveness of firewall relies on the security policy
  - **who** is allowed to talk to **whom**
  - **which** services are allowed to be used
- distinguish between inbound and outbound connections
  - **inbound**: attempts by external users to connect to services on internal machines
  - **outbound**: attempts by internal users to connect to services on external machines
- firewall more like the moat around the castle
  - control access to specific things

Firewall Locations

- between internal LAN and external network
- at gateways of sensitive subnetworks within LAN
  - e.g., payroll's network separated within corporate network
- on end-user machines
  - "personal firewall"
  - runs in OS of each machine
  - can provide "application-specific" rules

Inbound and Outbound

- **threat model** may suggest that inbound connections are riskier
  - internal users are authenticated
    - e.g., by logging into a computer
    - e.g., by having physical access
  - external users can be anyone on the internet
- example security policy
  - internal users can connect to any service
  - external users are restricted
    - **permit** connections to www service on port 80 and 443
    - **deny** connections to printer service port 631

Default Policy

- policy may specify permit and deny for different machines
- but how to treat traffic not mentioned in policy?
  - **default allow**
    - permit external access to services
    - shut off access as problems are seen
  - **default deny**
    - deny everything except specific things needed
    - e.g., ssh, web, etc.
    - add more when users complain
    - audit and approve changes

Default Policy

- which does design principles recommend?
- which notices flaws faster and with less risk?
- balance and consequence of false positives and false negatives
  - always relevant for imperfect binary decision making

Packet Filters

- most basic kind of firewall is a **packet filter**
  - a router with a list of **access control rules**
  - checks each received packet against the rules to decide what to do
    - forward to correct host
    - drop the packet entirely
  - each rule specifies which packets it applies to based on packet's header
    - is stateless, only considers the packet as is
    - use source / dest IP, ports, protocol names to judge
    - use * as a wildcard to match everything

Packet Filters

- stateless
  - cannot examine packet's context
  - e.g., payload information
- filtering rules based on header
  - pattern matching packet header fields
  - e.g, IP addresses, ports
  - e.g., TCP flags

Packet Filters Example

- allow tcp 1.2.3.4:1025 -> 10.0.0.1:80
  - firewall permits any TCP packet if
    - it is from 1.2.3.4
    - it is to 10.0.0.1
    - it is from port 1025
    - it is to port 80
- allow tcp 1.2.3.4:* -> 10.0.0.1:80
  - same as above but any source port okay

Packet Filter Examples

- rules can be ordered
  - first rule that applies decides
- examples
  - second rule inconsequential
    - deny tcp 1.2.3.4:* -> 10.0.0.1:*
    - allow tcp 1.2.3.4:* -> 10.0.0.1:80
  - allows port 80, disallows all other ports
    - allow tcp 1.2.3.4:* -> 10.0.0.1:80
    - deny tcp 1.2.3.4:* -> 10.0.0.1:*
  - deny tcp *:* -> *:*
    - default block everything
    - put this rule last and make everything before exceptions

Despite their simplicity this remains
the defacto way of running a firewall.

Despite their simplicity this remains
the defacto way of running a firewall.
Linux tool `iptables` is inspiration

Despite their simplicity this remains
the defacto way of running a firewall.
Linux tool `iptables` is inspiration
Extra features for convenience such as
port ranges, other headers, IP groups

FTP Example

- File Transfer Protocol (FTP)
  - client:2352 -> server:21 “use port 3573”
    - 2352 and 3573 are random ports
  - server:21 -> client:2352 “OK”
  - server:20 -> client:3573 “data...”
- this is a problem for firewalls
  - rules like allowing ports greater than 1023 used
  - but not ideal

Firewall Weakness

- do not prevent application-specific attacks
  - if a web server has a vulnerability, firewall will not block attack string
- no authentication
  - claimed source IP addresses can be spoofed
  - ingress filtering to ensure they are reasonable
    - i.e., list IP addresses for each interface
  - spoofed internal address can be detected
- vulnerable to misconfiguration
  - firewalls can have thousands of filtering rules
  - easy to introduce subtle errors
  - these need to be tested with unit tests like a program
    - e.g., lists to allow and deny and check

Firewall Weakness

- stateless filtering is not enough
  - servers run on well known ports, like 20, 21, 25, 80, 443
  - clients use random ports > 1023
    - these must be opened to allow responses
    - but any user-space network program can listen on these ports
  - allowing all connections above 1023 is insecure
  - must maintain state based on outgoing requests that are allowed to introduce temporary rules

Bastion Host

Bastion Host

- your public presence on the Internet
  - a machine that can be directly accessed
- analogy: lobby of a building
  - visitors may or may not be allowed upstairs
- bastion host must be kept secure
  - devote particular attention

Bastion Principles

- keep it simple
  - any software running can have vulnerabilities
  - only run what is needed
- assume it will be compromised
  - most likely machine to be attacked
  - most likely machine to attack your internal network
  - limit Bastion host's access to machines

Bastion Principles

- disable user accounts
  - normal users should not have accounts that allow them access to bastion host
    - vulnerability to attacker (including services for accounts)
    - harder to detect attacks
    - inadvertent subversion of the host's security by users

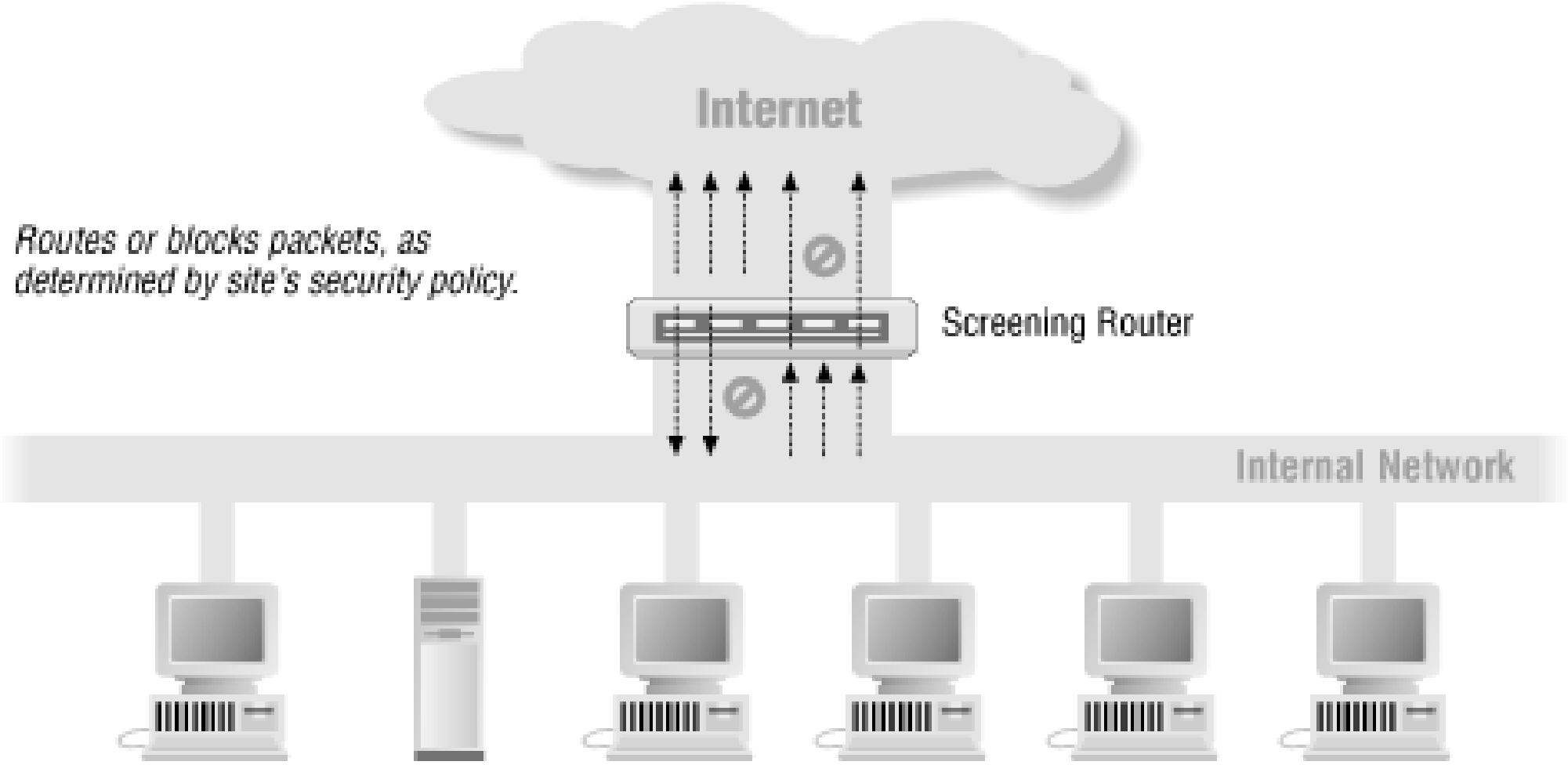
Operating a Bastion

- learn normal usage
  - how many jobs running
  - what CPU and memory usage
  - typical load at different times
- watch for reboots
  - crashes and reboots should be rare
  - should run for months without issue
  - crashes and reboots can be sign of an attack
- do secure backups
  - assume it will be compromised and needs to be rebuilt
  - can find evidence of attack by looking at current and old state
  - keep historical versions

Firewall Layouts

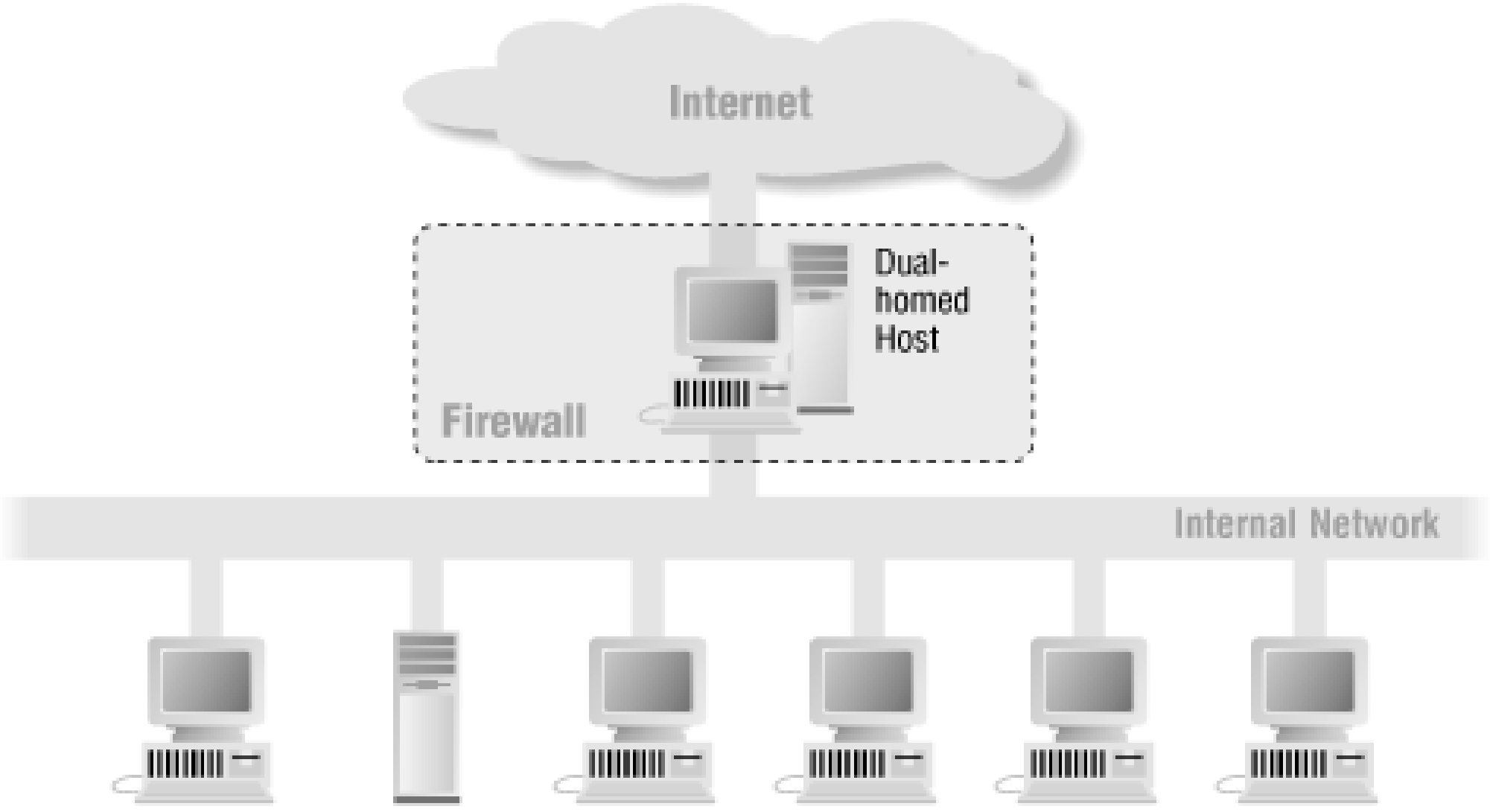
Screening Router

- a router between Internet and machines
- packet filter rules to arbitrate allowed traffic
- low-cost system
  - need a router anyways
- no defense in depth
  - compromised router is a compromised network
- probably what you have at home


Internet
Routes or blocks packets, as determined by site's security policy.
Screening Router
Internal Network

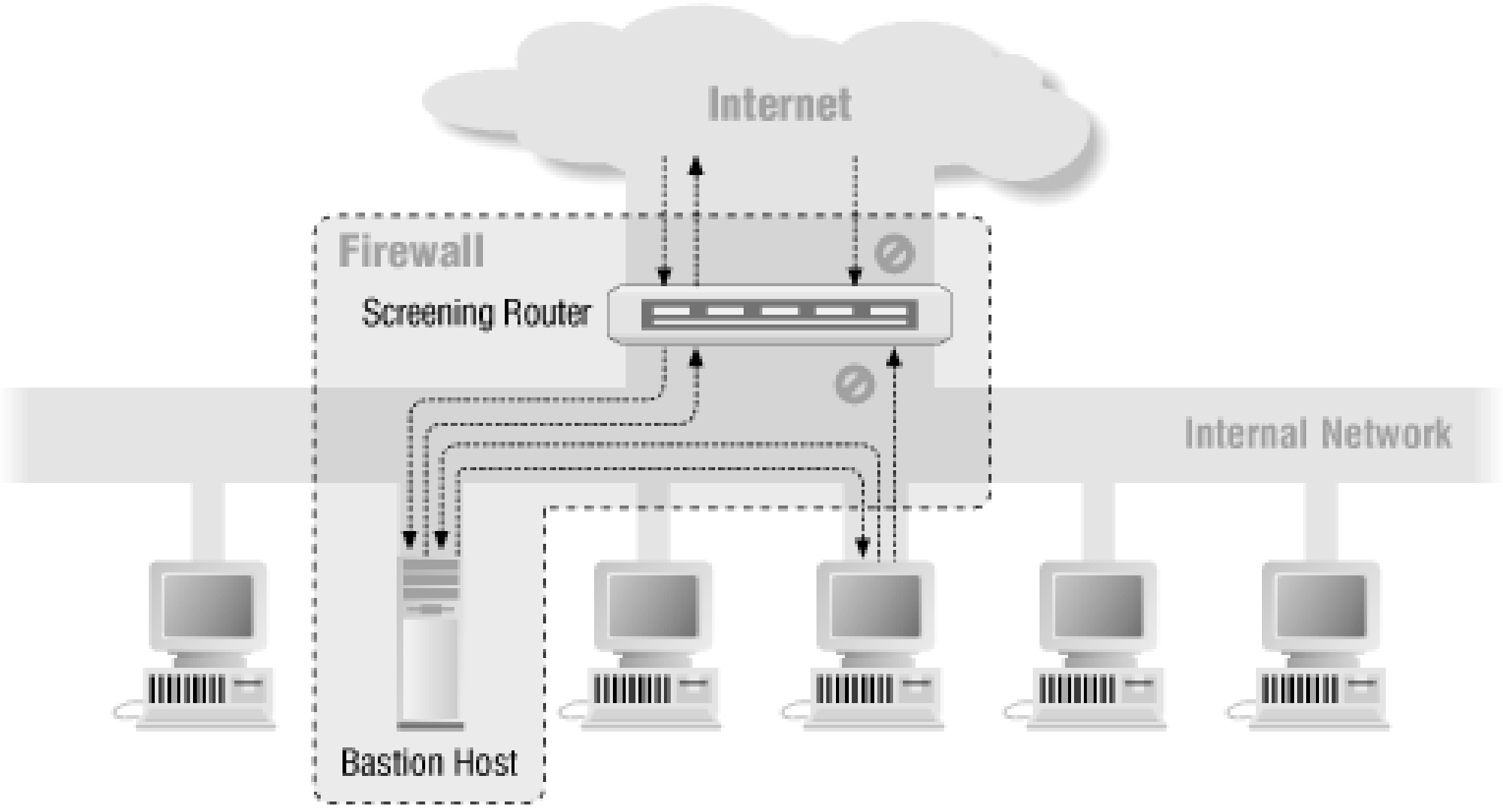
Dual-Homed Host

- a host computer that has two network interfaces
  - one for internet and one for internal network
  - host is thus a bastion host
- acts as a router between these networks
  - can run more complicated analysis
- no systems on Internet can communicate with any internal machine
  - there is no connection, network interfaces are separate
  - dual-homed host must mediate all communication


Internet
Dual-
homed
Host
Firewall
Internal Network

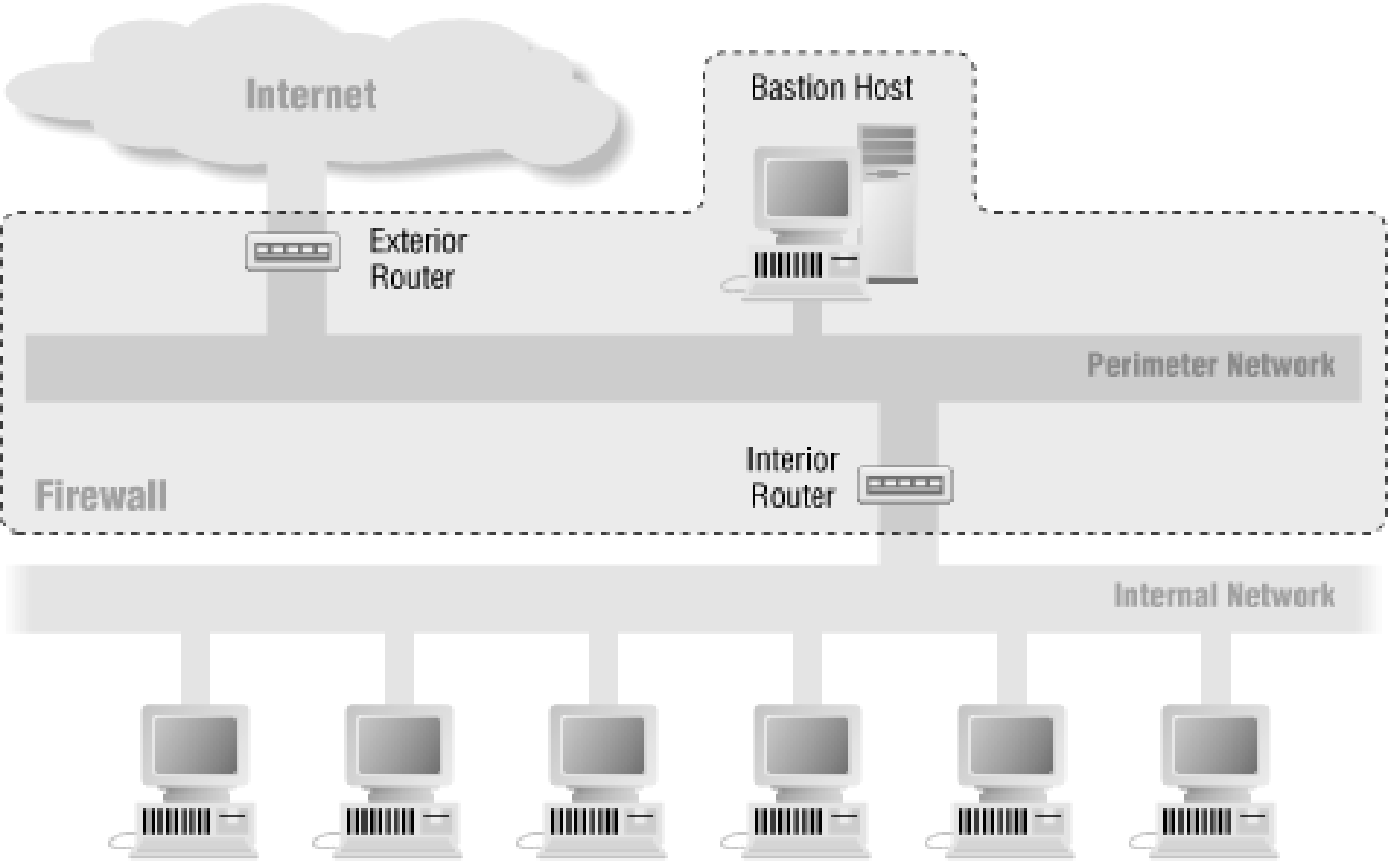
Screened Host

- a router screens traffic and only allows to particular bastion hosts
- bastion hosts are on the same internal network as other machines
- bastion host can, e.g., receive mail from the Internet
  - local machines can retrieve mail from bastion host
- not appropriate if bastion host serves many connections
  - e.g., acts as a webserver, file server, etc.


Internet
Firewall
Screening Router
Internal Network
Bastion Host

Screened Subnet

- adds a "perimeter network" or de-militarized zone (DMZ)
  - puts bastion hosts in the DMZ
  - has an "external" screening router between the Internet and the DMZ
  - has an "internal" screening router between the DMZ and the internal network
- allows bastion hosts to offer services to Internet
  - assumes they may be compromised
  - router protects internal machines
- different screening policies between the external and internal routers


Internet
Bastion Host
Exterior
Router
Perimeter Network
Interior
Router
Firewall
Internal Network

Screened Subnet

- perimeter network gives additional layer of security
  - attacker who breaches perimeter is still prevented from internal network
- a compromised machine may be able to examine all traffic on the network
  - e.g., Ethernet based networks
  - this limits the snooping to the perimeter
  - i.e., not between two internal network machines that may not be using encryption

Screened Subnet

- interior router
  - also called choke router
    - control point of monitor, control, and packet rejection
  - protects internal network from both internet and perimeter
    - can replicate rules that the external router should have implemented
    - can ensure no data is passing through that is internal-traffic only
  - limit services from bastion host to internal network as much as possible
    - e.g., DNS, SMTP

Screened Subnet

- exterior router
  - also called access router
  - protects perimeter net and internal net from Internet
  - generally allow almost anything outbound from perimeter
    - little filtering, not complicated
  - users may have such a router and not realize
    - e.g., provided by your ISP

Screened Subnet

- exterior router
  - one thing it can do is check forged senders
  - a packet "to" interior or perimeter "from" either cannot pass inbound on external router
  - prevents many types of attacks
  - impossible to tell afterwards
    - e.g., internal router cannot detect a spoofed packet "from" perimeter

A Screened Subnet is
almost always appropriate.

Variations

Variations

it's OK to use multiple bastion hosts

Variations

it's OK to use multiple bastion hosts

e.g., for redundancy, performance, separation of services, etc.

Variations

Variations

it's OK to merge the interior router and the exterior router

Variations

it's OK to merge the interior router and the exterior router provided you have multiple interfaces and can specify different rules for them.

Variations

Variations
it's OK to merge the bastion
host and the exterior router.

Variations

it's OK to merge the bastion host and the exterior router.

routers tend to offer better performance and configurations

Variations

it's OK to merge the bastion host and the exterior router.

routers tend to offer better performance and configurations

security concern is not large because filtering role of external router is minimal.

Non-Variations

Non-Variations
DO NOT merge interior router
and the bastion host.

Non-Variations

DO NOT merge interior router and the bastion host.

Bastion host and interior router perform different security roles that are not complementary.

Non-Variations

Non-Variations

DO NOT use multiple interior routers.

Non-Variations

DO NOT use multiple interior routers.
fundamentally, such a router might send interior traffic through the perimeter where it can be snooped on if someone broke past the bastion host.

Non-Variations

DO NOT use multiple interior routers.

fundamentally, such a router might send interior traffic through the perimeter where it can be snooped on if someone broke past the bastion host.

e.g., because it is a faster route

Non-Variations

DO NOT use multiple interior routers.

fundamentally, such a router might send interior traffic through the perimeter where it can be snooped on if someone broke past the bastion host.

e.g., because it is a faster route

it also requires consistent configs

Variations

Variations

it's OK to use multiple exterior routers

Variations

it's OK to use multiple exterior routers

e.g., because you have multiple connections

Variations

it's OK to use multiple exterior routers

e.g., because you have multiple connections

the threat of compromise increases

but we assume such devices are exposed.

Non-Variations

Non-Variations

DO NOT use screened subnet but also allow connections from the Internet directly to your internal network.

Non-Variations

DO NOT use screened subnet but also allow connections from the Internet directly to your internal network.

Seems obvious but sometimes (dangerous) exceptions are granted.

Non-Variations

DO NOT use screened subnet but also allow connections from the Internet directly to your internal network.

Seems obvious but sometimes (dangerous) exceptions are granted.

E.g., because a service needs to be granted but it is not just made into a bastion host on perimeter.

Why Have Firewalls Been Successful?

- central control
  - easy administration and update
  - single point of control
    - one config file to change
    - rapid response after changing
- easy to deploy
  - transparent to end users
  - simply add a device on the network that sits in front of the Internet
- addresses problem
  - security vulnerabilities in network services are rampant
  - easier to disable access to them than to secure them
  - easier to disable access if a new vulnerability appears

Firewalls Disadvantages

- functionality loss
  - some network stuff may not work
  - some apps don't work with both endpoints behind firewalls
- **insider** threat
  - firewalls assume that insiders are trusted
    - inbound versus outbound
  - this may not be the case
  - firewalls create a **security perimeter**
    - threats can come from laptops and cell phones that are compromised

Circumventing Firewalls

- packet filters have a **limited contextual model**
  - they look at headers
    - network and transport layer
  - they don't look at packets
    - application layer
  - using port 22 for ssh is a convention, not a rule
    - nothing preventing it on another port
    - ports like udp/53 for DNS are hard to block
- encrypted content is the norm
  - cannot be intercepted by a router without strenuous effort

Circumvention Technique: Abuse Ports

- port 53/udp is for DNS
  - typically this has to be allowed for the Internet to work
  - but why can't it be BitTorrent traffic instead?
    - provided client and server agree
    - port numbers are just a convention, not a rule
- how to get remote service to agree?
  - you could ask them to run it on a different port
  - you can run your own service and have it forward
    - "IP-over-DNS"

Circumvention with a Relay

- user runs a **relay**
  - a program listening on a port that is not blocked
  - e.g., HTTP
  - this program is running on a different network that is not behind a firewall
- user sends innocuous-looking traffic to their relay
- the traffic says "send the rest of the packet to IP:port"
- relay relays the traffic to the intended destination
- relay sends the reply back to the user
- how can this be detected?