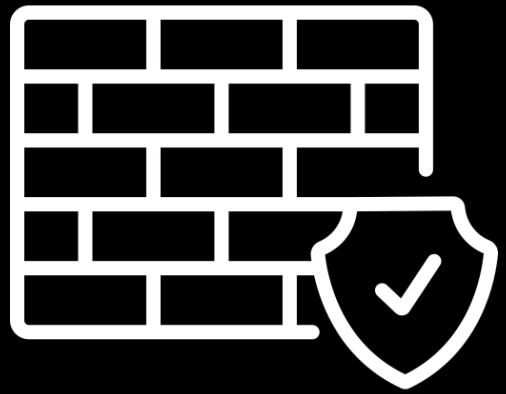

Firewalls

Lecture 08

Software Security Engineering

Winter 2023
Thompson Rivers University

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

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

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 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
- example: second rule inconsequential
  - deny tcp 1.2.3.4:* -> 10.0.0.1:*
  - allow tcp 1.2.3.4:* -> 10.0.0.1:80
- example: 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:*

How would you implement the default-deny rule?

How would you implement the default-allow rule?

What would it look like and where would you put it (relative to other rules)?

Firewall Considerations

- firewalls can have thousands of filtering rules
  - easy to introduce subtle errors
  - these need to be tested with unit tests like a program
- provides not only inbound security but outbound policy enforcement
  - e.g., disallows bittorrent on the network
- firewalls permit connections to be opened
  - internal:43256 -> external.com:443 thereafter allows reverse traffic

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
- suppose an internal user wanted to get around firewall
  - e.g., access forbidden content or use forbidden services
  - how may they do that?

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?