# Firewalls

## Overview

- In old days, brick walls (called **firewalls** ) built between buildings to prevent fire spreading from building to another
- Today, when private network (i.e., intranet) connected to public network (i.e., Internet), users communicate with outside world, and outside world with private network and its computer systems
- Intermediate system(s) placed between private network and public network to establish a controlled link, and a security wall or **perimeter** providing single point where security and audit may be imposed
- These intermediate systems called **firewall systems** or **firewalls** (alternative terms comprise **security gateways** and **secure Internet gateways**)

# Overview

- According to RFC 2828, term **firewall** refers to inter-network gateway that restricts data communication traffic to and from one of the connected networks, protecting that network's system resources against threats from other network
- It should have following properties
  - All traffic from inside to outside, and vice versa, must pass through the firewall
  - Only authorized traffic, as defined by local security policy, will be allowed to pass
  - Firewall itself immune to penetration (use of trusted system with secure operating system)

# Benefits and Limitations

- Pros
  - controlled and logged interaction with external Internet; can enforce security policy
  - internal machines can be administered with varying degrees of care
  - Focal point for security decisions
- Cons
  - services through firewall introduce vulnerabilities
  - performance may suffer
  - single point of failure
  - useless against insider attacks

# Firewall Characteristics

Four general techniques:

- Service control
  - Determines the types of Internet services that can be accessed, inbound or outbound
- Direction control
  - Determines the direction in which particular service requests are allowed to flow
- User control
  - Controls access to a service according to which user is attempting to access it
- Behavior control
  - Controls how particular services are used (e.g. filter e-mail)

# Firewall Evaluation Criteria 1

- **Performance**: Firewalls always impact performance - compare delays with respect to functions offered.
  - Authentication of connections
- **Requirements Support**: Should support the applications that are to be used across the network
  - SMTP, TELNET, FTP, HTTP, etc.

# Firewall Evaluation Criteria 2

- ***Access Control***: handled with IP addresses or user-based? How many users can be supported?
- ***Authentication***: How hard is this to administrate and how is it accomplished? Inbound and outbound?
- ***Auditing***: What gets audited? any audit reduction tools available?
- ***Logging/Alarms***: How is this accomplished? How is administrator notified?

# Firewall Evaluation Criteria 3

- ***Customer Support***: Training courses, installation, help desk, 24x7 availability?
- **Damage**: if compromised or destroyed, what outside threats can interfere with the 'protected' network, and how easy is this to detect and diagnose?
- **Physical Security Requirements**: Location requirements

## Firewall Design Philosophies

**Default deny**:

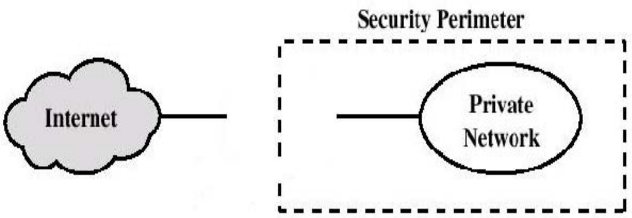
- *Everything not expressly permitted is prohibited*
  - Firewall designed to block everything
  - Services enabled case-by-case after careful analysis
  - Users more restricted and cannot easily breach security

**Default permit**:

- *Everything not expressly prohibited is permitted*
  - System administrator reacts to threats as discovered
  - Services are removed/limited when proven dangerous
  - Users are less restricted

## Types of Firewalls

- Ultimate Firewall

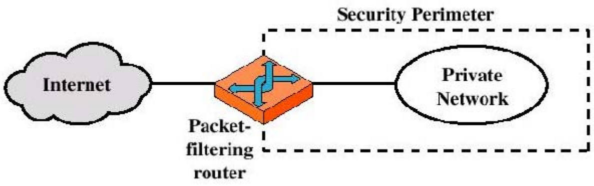


# Components

- Firewall policy
  - Service access policy
  - Firewall design policy
- Packet filters
  - Statically (stateless) filtering devices
  - Dynamically (stateful) filtering devices
- Application gateways
  - Circuit-level gateways
  - Application-level gateways or proxy servers

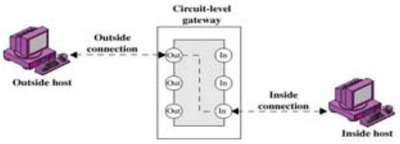
# Types of Firewalls

- Packet-filtering Router

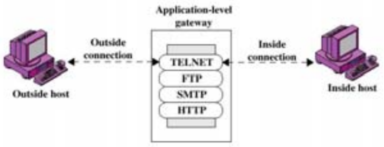


# Types of Firewalls

- Circuit-level Gateway



- Application-level Gateway



# Packet-filtering Router

- Security function consists of filtering (forward or drop) packet based on transport-layer information only
- These routers are sometimes called **screening routers**
- The following fields (usually) taken into account by any packet-filtering device
  - Network interface
  - IP header: Source address, Destination address
  - TCP or UDP header: Source and Destination ports
  - TCP connection flags (SYN,ACK,FIN, ...)
  - ICMP messsage type

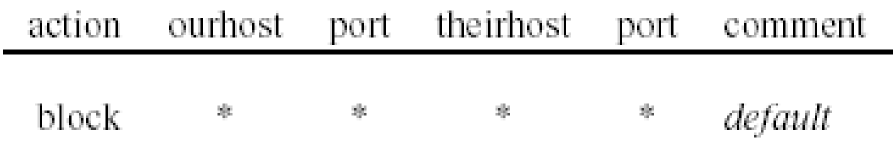
## Packet Filtering

- Advantages:
  - Simplicity
  - Transparency to users
  - High speed
- Disadvantages:
  - Difficulty of setting up packet filter rules
  - Lack of Authentication
  - Protect against amateur hackers only

## 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 the 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.

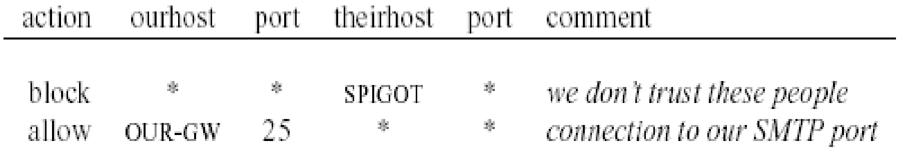


| action | ourhost | port | theirhost | port | comment |
|---|---|---|---|---|---|
| block | * | * | * | * | *default* |

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 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* |

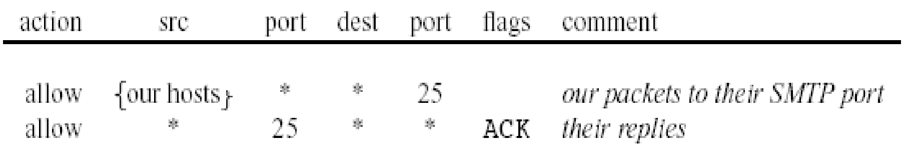
This solution allows calls to come from any port on an inside machine, and will direct them to port 25 on the outside. Simple enough…

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.

What can be a better solution ?

## better Solution 2:



| 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

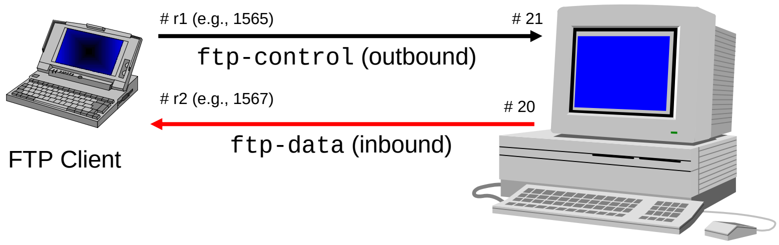
# Packet Filtering

- Order rules so that most common traffic is dealt with first
- Correctness is more important than speed

- Possible attacks
  - IP address spoofing
  - Source routing attacks
  - Tiny fragment attacks

# Packet Filtering

- A packet filter can be **stateless**, meaning that each IP packet is treated individually
- Practical problems occur if inbound connections must be established to dynamically assigned port numbers (e.g., FTP data connection): request may be rejected.


# r1 (e.g., 1565)

# 21

`ftp-control` (outbound)

# r2 (e.g., 1567)

# 20

`ftp-data` (inbound)

FTP Client

FTP Server

# Packet Filtering

- In case of FTP, **passive mode FTP** solves the problem, as FTP data connection is also established outbound (from client to server)
- Underlying problem is more general and applies to increasingly large number of applications (e.g., CORBA InternetInterOrbProtocol and many UDP-based and real-time application protocols)
- One way to address the problem is to have packet filters establish and maintain state information to more intelligently filter TCP connections or UDP datagram transport sessions

# 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

# Packet Filtering

- Stateful Packet Filtering
  - filters based on:
    - Information contained in the current packet
    - Information contained in previous packet transmitted
  - Accomplished using state table
    - Maintains state information about the communication from previous packet (client-server session)
  - Information comes from any part of the packet

## Packet Filtering (stateful)

- Advantages
  - Can deal with most of the problems that can rise from using stateless filtering
    - Can handle UDP packets
    - Can handle fragmented packets
    - Can prevent TCP Open SYN Flood Attacks
  - Disadvantages
    - Not easy to configure
    - Less secure than Application level gateways???

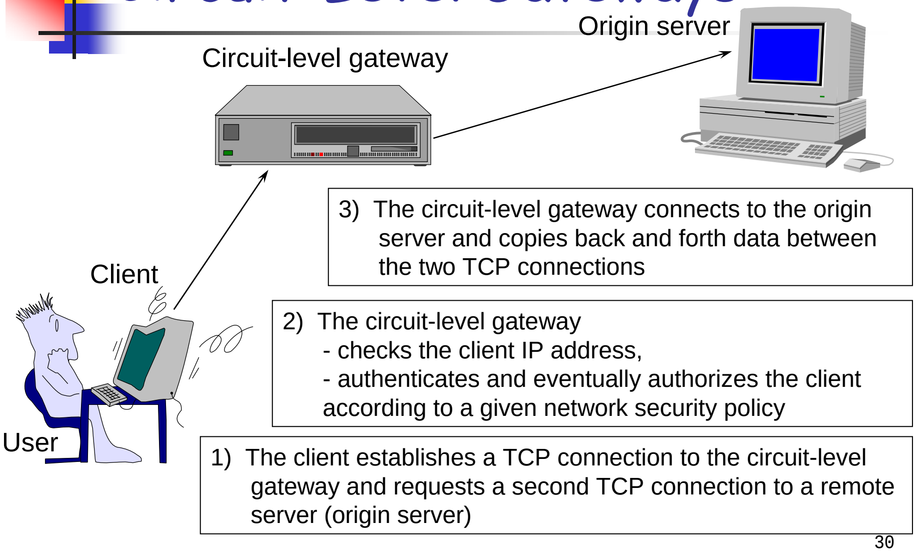
## Types of Firewalls

- **Circuit-level Gateway**
  - Stand-alone system or specialized function performed by an Application-level Gateway
  - Sets up two TCP connections
  - The security function consists of determining which connections will be allowed
  - The gateway typically relays TCP segments from one connection to the other without examining the contents
  - Typical use is a situation in which the system administrator trusts the internal users
  - An example is the SOCKS package

# Circuit-Level Gateways

- **A circuit-level gateway** is essentially a proxy server for transport layer associations (i.e., TCP connections) [although more recent ones can handle UDP-based application protocols]
- A circuit-level gateway differs from a port-forwarding mechanism
  - Unlike a port-forwarding mechanism, the client must be made aware of the circuit-level gateway
  - Contrary to a port-forwarding mechanism, the circuit-level gateway is generic in the sense that it can handle any TCP connection (if enabled in its configuration)

# Circuit-Level Gateways



3) The circuit-level gateway connects to the origin server and copies back and forth data between the two TCP connections

2) The circuit-level gateway
- checks the client IP address,
- authenticates and eventually authorizes the client according to a given network security policy

1) The client establishes a TCP connection to the circuit-level gateway and requests a second TCP connection to a remote server (origin server)

# Circuit-Level Gateways

- A common circuit-level gateway is **SOCKS** (Refer to `https://en.wikipedia.org/wiki/SOCKS`)
- Original implementation consisted of two components
  - **SOCKS server** or **daemon** (i.e., `sockd`)
  - **SOCKS library** used to replace regular Sockets calls in the client software
- The application developer has to recompile and link the client software with a few preprocessor directives to intercept and replace the regular TCP/IP networking Sockets calls with SOCKS counterparts (`connect` with `Rconnect`, `listen` with `Rlisten`, etc.)

# Circuit-Level Gateways

- The goal of SOCKS was to provide a general framework for TCP/IP applications to securely use (and traverse) a firewall
- When a client requires access to a server on the Internet, it must first open a TCP connection to the appropriate port (1080) on the SOCKS server residing on the firewall system. Then the client uses the SOCKS protocol to have the SOCKS server establish a second TCP connection to the origin server

# Types of Firewalls

- **Application-Level Gateways**
  - Acts as a relay of application-level traffic
    - Does not provide the service itself. It only acts as the client to the real server
  - It interprets the application protocol, and therefore checks or filter the content
  - works at the application layer, is specific and generally able to proxy only one TCP-based application protocol
  - A firewall needs specific application-level gateways (or **proxy servers**) for every application protocol that must traverse the firewall (a serious disadvantage for, e.g., proprietary protocols)

# Application-Level Gateways

- Advantages:
  - Higher security than packet filters
  - Only need to scrutinize a few allowable applications
  - Easy to log and audit all incoming traffic
- Disadvantages:
  - Additional processing overhead on each connection (gateway as splice point)

# Application-Level Gateways

- In general, the use of an application gateway requires some modification of either the user procedures or the client software (not convenient either way)
- Useful to have a firewall that maintains all software modifications required for application gateway support in the firewall
- Solution: **transparent firewalls,** configured to listen on the network segment of the firewall for outgoing TCP connections and to relay these connections on the behalf of the client.

# Application-Level Gateways

- Transparency is not necessarily provided in both directions (e.g., inbound transparency is seldom required or used)
- A transparent firewall requires that all messages to and from the Internet be transmitted through the firewall
- Similar functionality is required for **network address translation (NAT)**

# Application-Level Gateways

- The application-level gateway must be able to authenticate and authorize user requests
  - List of IP addresses that are allowed to connect inbound or outbound
  - Weak authentication schemes (e.g., password)
  - Strong authentication schemes
- In practice, the firewall policy must define the authentication and authorization schemes that must be used in either direction and for each service
- Many policies use the simplest scheme mentioned above for outbound connections and a strong authentication scheme for inbound connections

# Application-Level Gateways

- Need for access to reference information to verify the authentication of the information provided by the client (e.g., hash value of a user password or the public key certificate for a specific user)
- The reference information can be stored either locally or remotely: the latter is preferable since it makes it possible to aggregate at a single point security information for several firewall systems and network access servers

# Application-Level Gateways

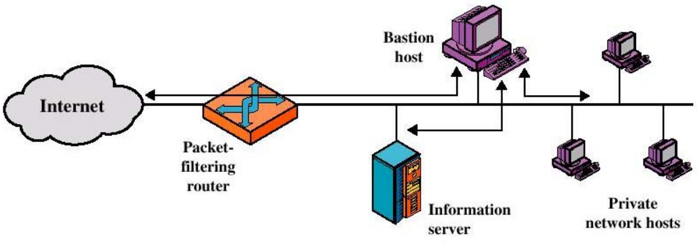
- A standardized protocol is used to retrieve the reference information from a centralized security server
- Protocols
  - **Remote Authentication Dial-In User Service (RADIUS)** developed by Livingston Enterprises, Inc.
  - **Terminal access controller access control system (TACACS)** now replaced by TACACS+ developed by Cisco Systems
- Both protocols widely supported by commercial firewall systems and network access servers

# Firewall Configurations

- More complex configurations than a simple system (single packet filtering router or single gateway) are possible.
- Three common configurations, all using the notion of **Bastion Host**
  - A system identified by the firewall administrator as a critical strong point in the network´s security
  - The bastion host serves as a platform for an application-level or circuit-level gateway

# Firewall Configurations

- Screened host firewall system (single-homed bastion host)



# Screened host single-homed

Single-homed bastion configuration

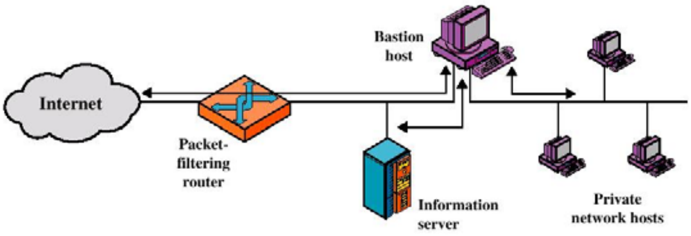
- Firewall consists of two systems:
  - A packet-filtering router
    - only packets from and to the bastion host are allowed to pass through the router
  - A bastion host
    - performs authentication and proxy functions

# Screened host single-homed

- Greater security than single configurations because:
  - This configuration implements both packet-level and application-level filtering (allowing for flexibility in defining security policy)
  - An intruder must generally penetrate two separate systems to compromise network
- This configuration also affords flexibility in providing direct Internet access (public information server, e.g. Web server) by allowing packets through

# Firewall Configurations

- Screened host firewall system (dual-homed bastion host)



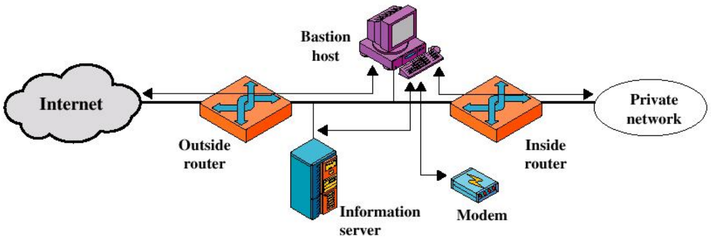
# Screened host dual-homed

Dual-homed bastion configuration

- In single-homed, if packet-filtering router is completely compromised, traffic flows directly to private network
- In dual-homed, traffic between the Internet and other hosts on the private network has to flow through the bastion host too

# Firewall Configurations

- Screened-subnet firewall system



## Screened subnet

- Most secure configuration of the three
  - Three levels of defense to thwart intruders
- Two packet-filtering routers are used
- Creation of an isolated sub-network
  - *Inside* router advertises only the existence of the screened subnet to the internal network (the systems on the inside network cannot construct direct routes to the Internet)
  - *Outside* router advertises only the existence of the screened subnet to the Internet (internal network is invisible to the Internet)

## Conclusions

If properly designed, implemented, deployed and administered, a firewall can provide effective access control services

The firewall technology is the most widely deployed security technology on the Internet

It cannot protect

- from attacks bypassing it, e.g., utility modems, trusted organisations, trusted services (SSL/SSH)
- against internal threats e.g., disgruntled employee
- against transfer of all virus-infected programs or files