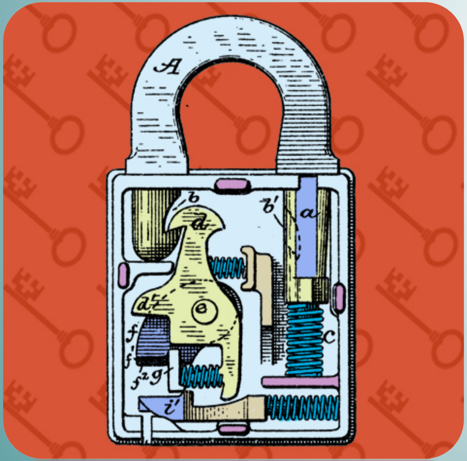
Garantía y Seguridad en Sistemas y Redes

Tema 10. Intrusion Prevention



Esteban Stafford

Departamento de Ingeniería
Informática y Electrónica

Contents

The Need for Firewalls

Firewall Characteristics

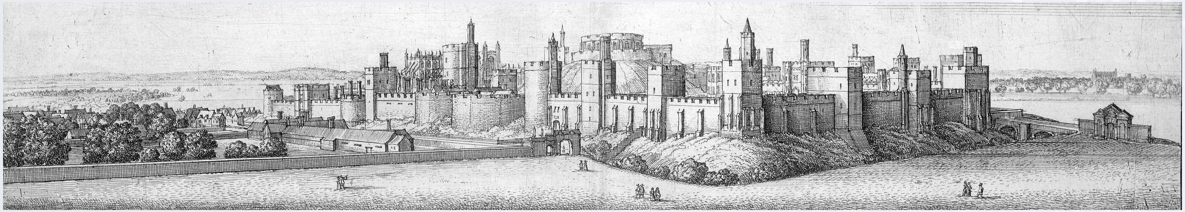
Types of Firewalls

Firewall Basing

Firewall Location and Configurations

Firewalls and Intrusion Prevention Systems

- Operating systems and applications are insecure.
- Internet connectivity is essential...
  - for every organization and individuals.
  - but it is a risky place. External threats.
- Firewalls set up a perimeter defense, giving:
  - “They” vs. “We”; Outside vs. Inside.
  - Single choke point to impose security on a LAN.
  - Auditing point for identifying problems afterwards.
- Firewalls are just another layer of defense.
- Military Doctrine: Defense in depth



Firewall Design

Design Goals

- Firewall is the **only way in or out** the perimeter.
- Only authorized traffic is allowed to pass.
- The firewall itself is immune to penetration.

Access control techniques

- **Service control**: which service is allowed (Port numbers...)
- **Direction control**: from outside/to inside access.
- **User control**: control access depending on user. Requires authentication.
- **Behavior control**: how particular services are used: spam filter on email or hide web for external users.

Firewall Capabilities and Limits

Capabilities

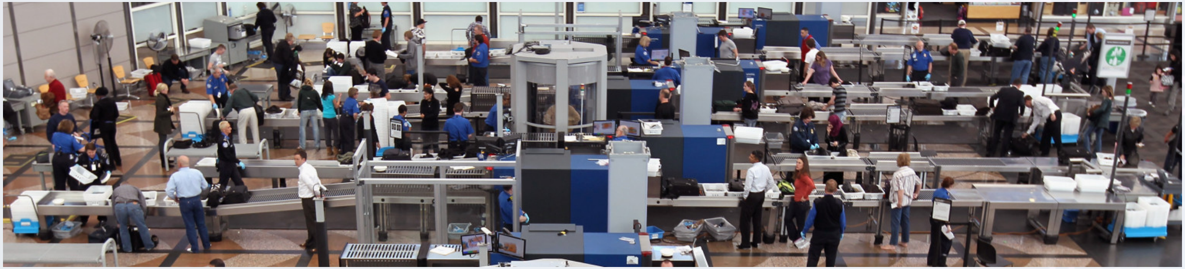
- Defines single entry point.
- Provides a location for monitoring security events.
- Convenient platform for some Internet functions
  - Routing, NAT, usage monitoring, IPSEC VPNs...

Limitations

- Cannot protect against attacks bypassing firewall
  - Dial-out, mobile broadband, WiFi.
- May not protect against internal threats.
- Laptops, PDA, portable storage device infected outside then used inside.

Types of Firewalls

- Packet filters.
  - Applies simple rules to allow or discard packets.
- Statefull packet filters.
  - Rules might involve previous history.
- Application firewalls.
  - Filters traffic attending to higher layer protocols.
- Proxies.
  - Allow communication on a connection basis.



Packet Filtering Firewall

- Fast and transparent to users.
- Applies simple rules to traffic through firewall.
- Based on information in packet header:
  - Src/dest IP addr and port, protocol, interface, TCP state...
- When a rule matches it applies an action:
  - Accept, drop, reject, log...
- If no rule matches, the it applies default policy:
  - Accept - permit unless expressly prohibited
  - Drop - prohibit unless expressly permitted

Source	Destination	Protocol	Port	Action
192.168.1.0/24	192.168.1.100	TCP	22	Accept
any	192.168.1.101	TCP	80	Accept
any	any	any	any	Drop

Packet Filter Weaknesses

Weaknesses

- Cannot prevent application level exploits.
- Limited logging functionality.
- Do not support advanced user authentication.
- Vulnerable to attacks on TCP/IP protocol bugs.
- Improper configuration can lead to breaches.
- Complex configurations end up with too many rules.

Attacks and countermeasures

- IP address spoofing: Discard external packets with internal addresses.
- Source-routing attacks: Discard packets with it.
- Tiny fragment attacks: Discard packets.

Stateful Inspection Firewall

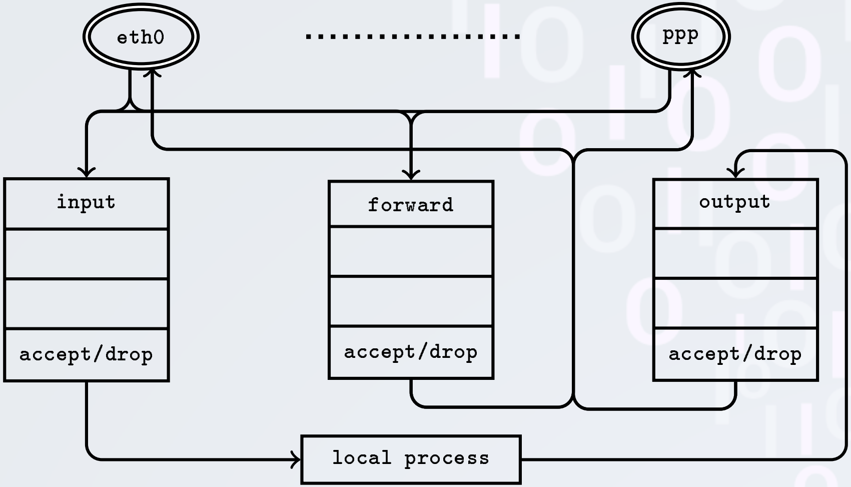
Motivation

- Packet filters have no memory (stateless).
- Complex protocols cannot be properly handled (TCP, FTP...)
- Based on past information better filters can be built.

Capabilities

- Review packet header information
- But also keep track of connections and other information.
- Can be used to close unused inbound high ports (TCP).
- Can track sequence numbers (Prevent session hijacking).
- Can make simple checks on higher level protocols (FTP, IM).

iptables



- Input chain: Filters traffic that will be consumed by local processes.
- Forward chain: Filters traffic routed to other hosts.
- Output chain: Filters traffic comming from local processes.

iptables

- No rules + policy ACCEPT = no firewall

```
$ iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source    destination

Chain FORWARD (policy ACCEPT)
target     prot opt source    destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source    destination
```

- Change policy

```
$ iptables -P INPUT DROP
```

iptables

- Add simple rule

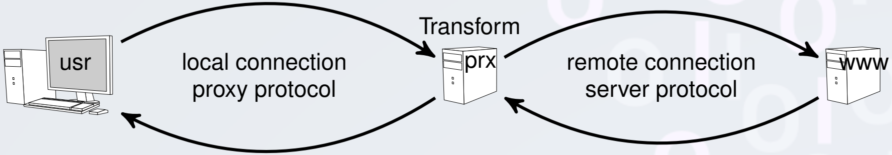
```
$ iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

- Only accept HTTP

```
Chain INPUT (policy DROP)
target     prot opt source    destination
ACCEPT     tcp  --  anywhere  anywhere     tcp dpt:http
```

Application-Level Gateway

- Also known as **Application Proxy**.
- A proxy is a connection relay.



- Recognizes application-specific commands and offers security controls.
  - Can perform user authentication.
  - May restrict application features supported.
  - Deep packet-inspection: can make serious checks.
- Not always transparent. Applications need to know about the proxy.
- Impose a higher overhead on traffic management.

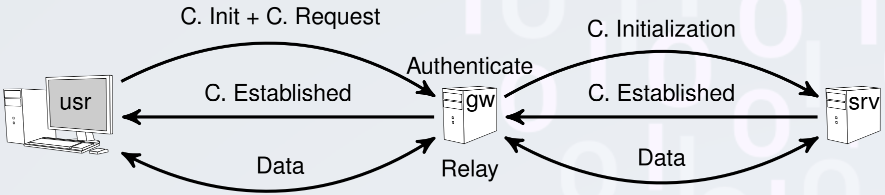
HTTP Proxy

```
GET / HTTP/1.1
User-Agent: Wget/ 1.13.4 (linux-gnu)
Accept: */*
Host: www.google.es
Connection: Keep-Alive
```

```
GET http://www.google.com/ HTTP/1.1
User-Agent: Wget/1.13.4 (linux-gnu)
Accept: */*
Host: www.google.com
Connection: Close
Proxy-Connection: Keep-Alive
```

Circuit-Level Gateway

- Similar to a proxy, but for any tcp connection.



- Relays TCP segments from one connection to the other without examining contents. Proxies translate between local and remote protocols.
  - Hence independent of application logic.
  - Just determines whether relay is permitted.
- Typically used when inside users trusted:
  - May use application-level gateway inbound connections
  - And circuit-level gateway outbound connections.
  - Hence lower overheads.
- SOCKS (RFC1928) allow TCP/UDP applications to securely use firewall

Bastion Hosts

- Critical strongpoint in network.
- Usually hosts application/circuit-level gateways.
- Common characteristics:
  - Runs secure O/S, only essential services (No login).
  - May require user authentication to access a proxy.
  - Each proxy can restrict features/hosts accessed.
  - Each proxy small, simple, checked for security.
  - Each proxy is independent, non-privileged (Jail).
  - Limited disk use, hence read-only code.

Host-Based Firewalls

- A module to secure individual hosts.
  - Available in many O/S: Linux iptables
  - Or an add-on module.
- Similar to standard firewall to filter packet flows.
- Often used on servers
- Advantages:
  - Tailored filter rules for the specific host needs.
  - Protection from both internal/external attacks
  - Another layer of protection, additional to network firewall.
  - Another layer of complexity, really necessary?

Screening Router

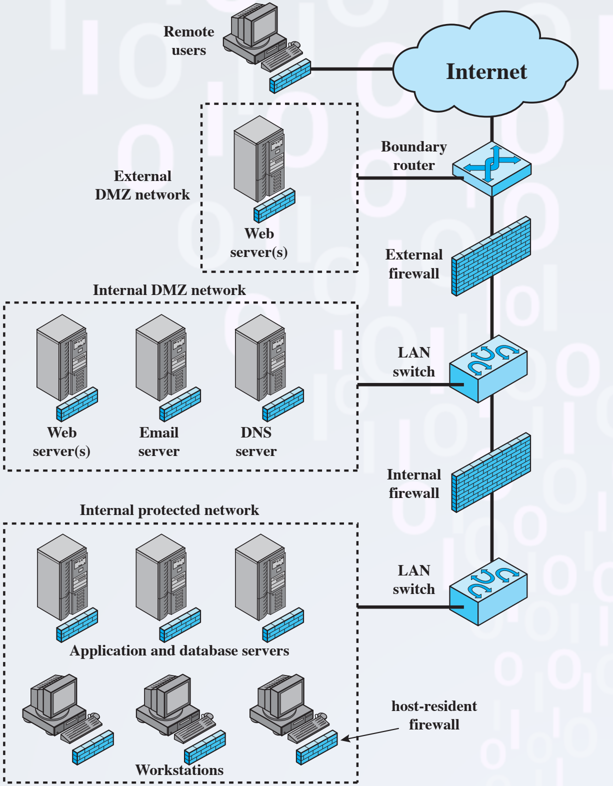
- On home cable/DSL router/gateway
- For both home or corporate use
- Typically much less complex.
- Primary role to deny unauthorized access.
- May also monitor outgoing traffic to detect/block malware activity.
- Potential problems:
  - Block some applications or services which are not sepecifically allowed by the firewall.

Firewall Topologies

- Host-resident firewall.
- Screening router (Home ADSL).
  - Packet filtering.
- Single bastion inline firewall.
  - Like screening router with more sofisticated firewalls.
- Single bastion T:
  - Inside vs. outside vs. DMZ.
  - Has a third network interface.
- Double bastion inline.
  - DMZ Between two firewalls.
- Double bastion T: outside, internal servers, users.
- Distributed firewall configuration.

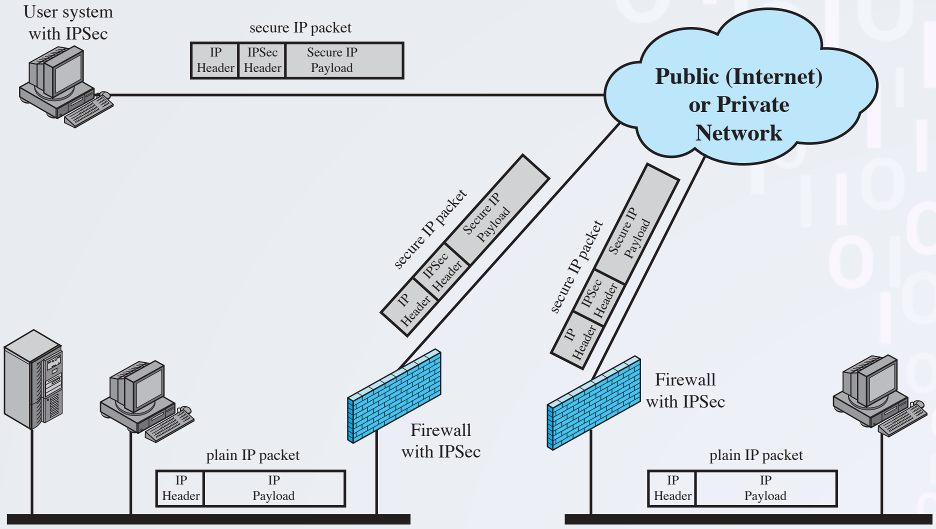
Distributed Firewalls

- A central control + Standalone firewalls + host-based firewalls.
- Comprehensive controls allow finer granularity.
- Internal DMZ
- External DMZ



Virtual Private Networks (VPNs)

- VPNs are a cheap way of implementing distributed internal network.
- IPsec: uses encryption and authentication in the network layer to provide a secure connection.



User system with IPSec
secure IP packet
IP Header
IPSec Header
Secure IP Payload
Public (Internet) or Private Network
secure IP packet
IP Header
IPSec Header
Secure IP Payload
secure IP packet
IP Header
IPSec Header
Secure IP Payload
Firewall with IPSec
Firewall with IPSec
plain IP packet
IP Header
IP Payload
plain IP packet
IP Header
IP Payload