# Network Perimeter Defense

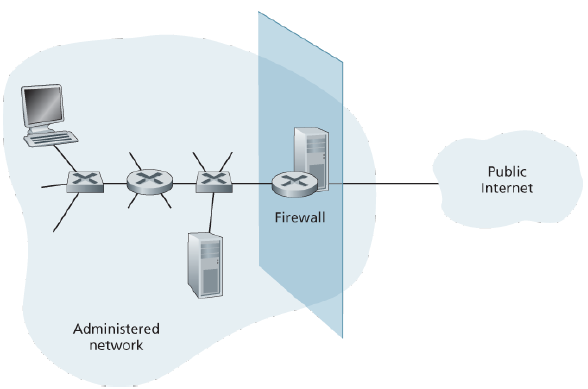
## Securing Network Perimeter



* Think of it like fortifying a castle:
  + Moat surrounding the castle
  + Limited entry points to minimize break-in
  + Thick outer and inner walls as barriers
  + Watch towers and guards at gate to identify enemies
* Basic access points into a network
  + LANs inside the organization (public facing services, e.g. servers)
  + Facilitated via Internet (most attacks come in this way)
* Basic elements in preventing access
  + Firewalls
  + Intrusion detection systems (IDS)
  + Network Address Translation (NAT)

## Firewalls

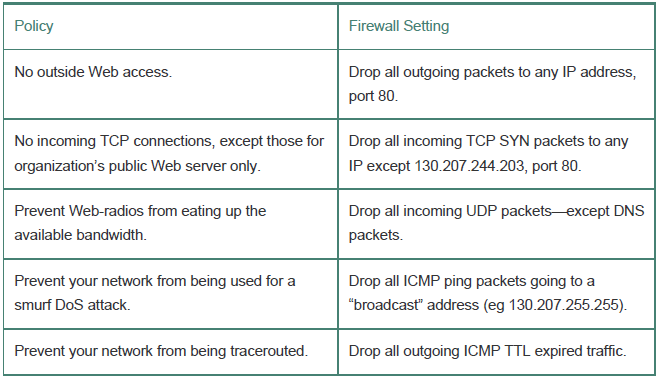
* firewall = combination of hardware + software that isolates an organization’s internal network from the Internet at large
  + allows only trusted packets to pass
  + blocks others
* firewall allows for net admin to control access between outside world and resources within admin’s net by managing traffic flow to and from these resources.
* 3 types of firewalls: traditional packet filters, stateful filters, application gateways.



## Firewalls: Why?

* A firewall has the following goals:
  + allow only authorized access to inside network
    - set of authenticated users/host
  + prevent denial of service attacks
    - e.g. prevent SYN flooding (where attacker establishes many bogus TCP connections to crowd out real connections)
  + prevent illegal modification/access of internal data
    - e.g. stop attacker from replacing CIA’s homepage with something else
  + firewall itself is immune to penetration (but it may be brought down)

## Traditional/Stateless Packet Filtering

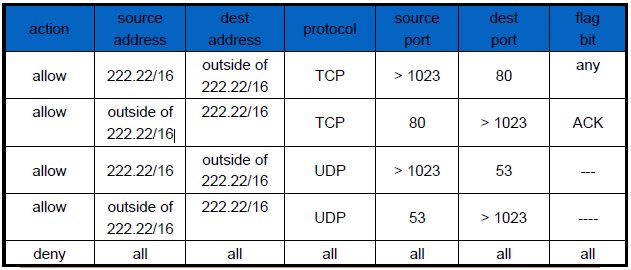
* internal network connected to Internet typically via gateway router (which also acts as firewall)
* router filters packet by packet, decision to forward/drop packet based on administrator specific rules and typically based on:
  + IP source or destination address
  + Protocol type in IP datagram field: TCP, UDP, ICMP, OSPF, and so on
  + TCP or UDP source and destination port
  + TCP flag bits: SYN, ACK, and so on
  + ICMP message type
  + Different rules for datagrams leaving and entering the network
  + Different rules for the different router interfaces
* One weakness: admits packets that “make no sense”; e.g. dest port 80, ACK bit set, but no TCP connection actually established (see ACL).

## Stateless packet filtering: example

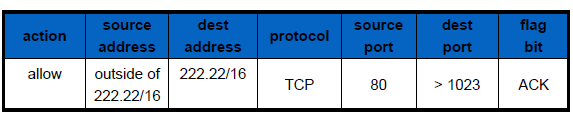
* example 1: block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23
  + result: all incoming, outgoing UDP flows and telnet connections are blocked
* example 2: block inbound TCP segments with ACK=0.
  + result: prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

## Access Control Lists

* ACL = table of rules (in action, condition pairs) that is applied top to bottom to incoming packets.



* stateless packet filter would not block this packet, even though it makes no sense (TCP connection not made).



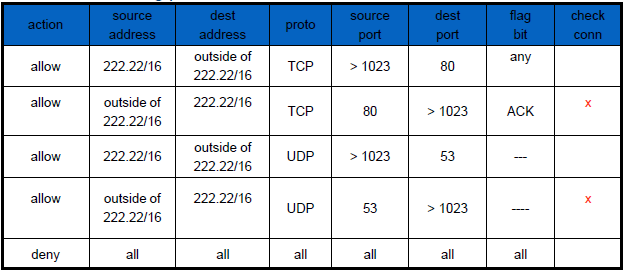
* however, stateful packet filter, which tracks status of every all TCP connections, would block it

## Stateful Packet Filters

* Stateful packets filters actually track TCP connections and use this knowledge to make filtering decisions, unlike stateless filters which do not check connections.
  + tracks status of every TCP connection
  + tracks connection setup (SYN), teardown (FIN)
  + determines whether incoming, outgoing packets “makes sense”
  + timeout inactive connections at firewall: no longer admit packets

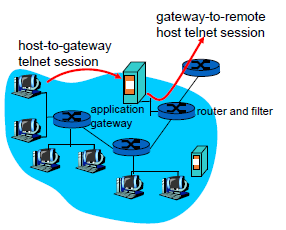
## Stateful Packet Filters: ACL Difference

* ACL augmented to indicate need to check connection state table before admitting packet



## Application Gateways

* filters packets on application data as well as on IP/TCP/UDP fields.
* example: allow select internal users to telnet outside.
  1. require all telnet users to telnet through gateway.
  2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
  3. router filter blocks all telnet connections not originating from gateway.



## Limitations of Firewalls, Gateways

* IP spoofing: router can’t know if data “really” comes from claimed source
* if multiple app’s need special treatment, each has own application gateway
* client software must know how to contact gateway.
  + e.g., must set IP address of proxy in Web browser
* filters often use all or nothing policy for UDP
* tradeoff: degree of communication with outside world, level of security
* many highly protected sites still suffer from attacks
  + firewalls can be taken down

# Intrusion Detection Systems

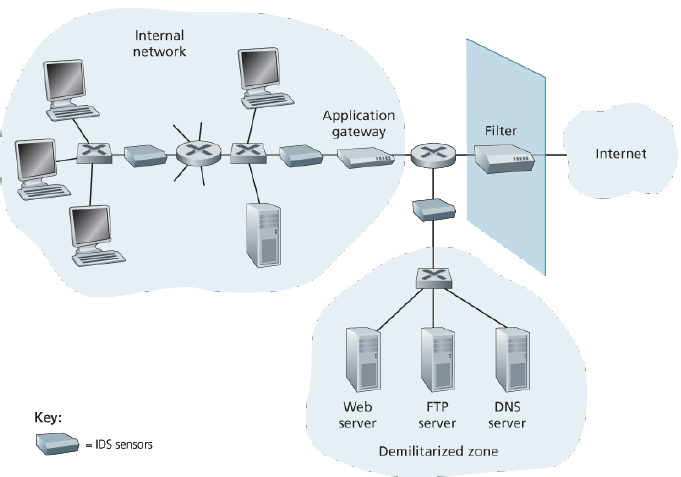
## Intrusion Detection Systems (IDS)

* packet filtering:
  + operates on TCP/IP headers only
  + no correlation check among sessions
  + little to no deep packet inspection (look at actual app data)
* intrusion detection system:
  + does header checking + deep packet inspection
    - deep packet inspection = look at packet contents (e.g. check char strings in packet against database of known virus strings)
  + examine correlation among multiple packets
  + generates alerts when it detects potentially malicious traffic
  + also filters out suspicious traffic (using intrusion prevention system).

  
Example: port scanning by Nmap

## More About IDS’s

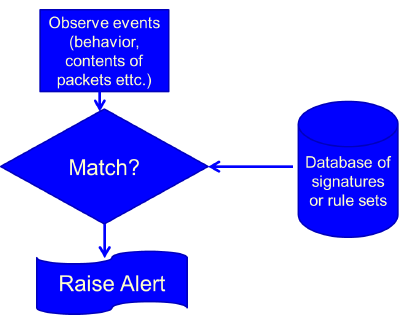
* An IDS can be used to detect a wide range of attacks
  + network mapping (attackers study the connectivity of networks)
  + port scans (attackers check which ports are open and could be exploited)
  + TCP stack scans
  + DoS bandwidth-flooding attacks,
  + worms and viruses
* A lot of today’s IDS systems are proprietary (marketed by Cisco, Check Point, others). However, others are public-domain systems, like Snort.
* Multiple IDSs can be deployed…
  + different types of checking at different locations
  + typically work in concert, sending info to central IDS processor
  + networks can be divided into higher-sec (protected by packet filters, app gateways, and IDS sensors) and lower-sec/DMZ zones (protected by packet filter only, but also monitored by IDS sensors)



* IDS systems broadly classified as either **signature-based** or **anomaly-based** systems.

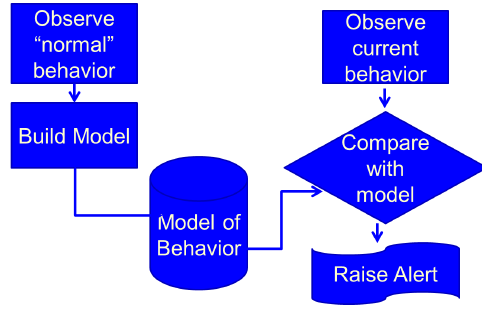
## Signature-Based/Misuse-Based IDS

* maintains an extensive database of attack signatures
  + signature = set of rules pertaining to an intrusion activity
  + signature may be a list of characteristics about a single packet (e.g., src and dest port #, protocol type, etc) or may relate to a series of packets.
  + signatures normally created by skilled network security engineers who research known attacks.
  + an org’s net admin can customize the signatures or add its own to the database.
* operationally, signature-based IDS…
  + sniffs every packet passing by it
  + compares each sniffed packet with the signatures in its database.
  + if a packet (or series of packets) matches a signature in the database, the IDS generates an alert.
* Have a number of limitations:
  + can only identify known attacks
  + needs regular updates
  + sometimes, false alarm is generated
  + can become overwhelmed with processing, thus fail to detect bad packets



## Anomaly-Based IDS

* creates a traffic profile of expected “normal” behavior.
  + attacks assumed to exhibit different pattern.
  + able to detect unknown attacks.
* Example of “normal”:
  + User logs on every weekday 9am.
  + Accesses supplier websites.
  + Logs off at 5pm.
* Example of “suspicious”:
  + User logs on at 3am.
  + Installs new software.
* Weakness – hard to distinguish between normal and unnatural behavior, so lots of potential for false alarms
* Anomaly-based systems rarely used; usually some anomaly-based features incorporated in signature-based system instead.



## Example: Snort

* Lightweight IDS system capable of performing real-time traffic analysis and packet logging
* Snort has three primary uses. It can be used as:
  + a packet sniffer like tcpdump
  + a packet logger (useful for network traffic debugging, etc)
  + a full network intrusion detection system



## Honeypots

* Computer or network appearing legitimate.
* Actually, a trap known as a honeypot.
* Used to study attacks or draw an attacker out.
* Monitor attacker behavior with no risk to real assets.

