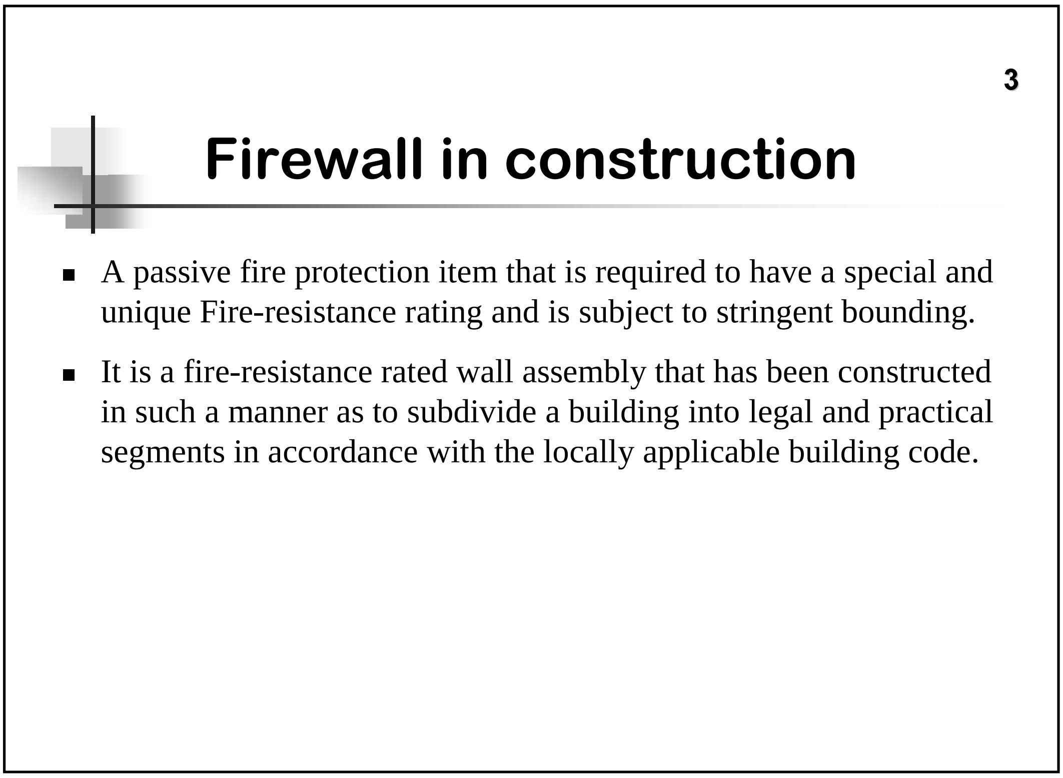
Firewalls

Dr.Talal Alkharobi

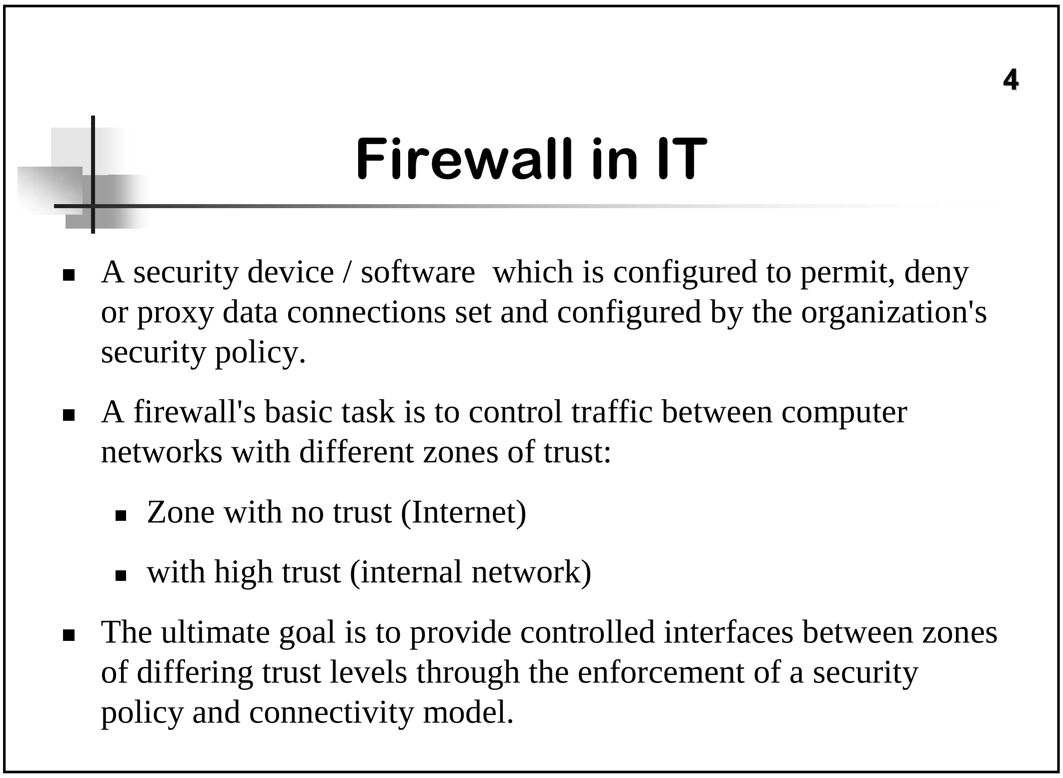
Castle and Moat Analogy

- More like the moat around a castle than a firewall
  - Restricts access from the outside
  - Restricts outbound connections, too (!!)
    - Important: filter out undesirable activity from internal hosts!



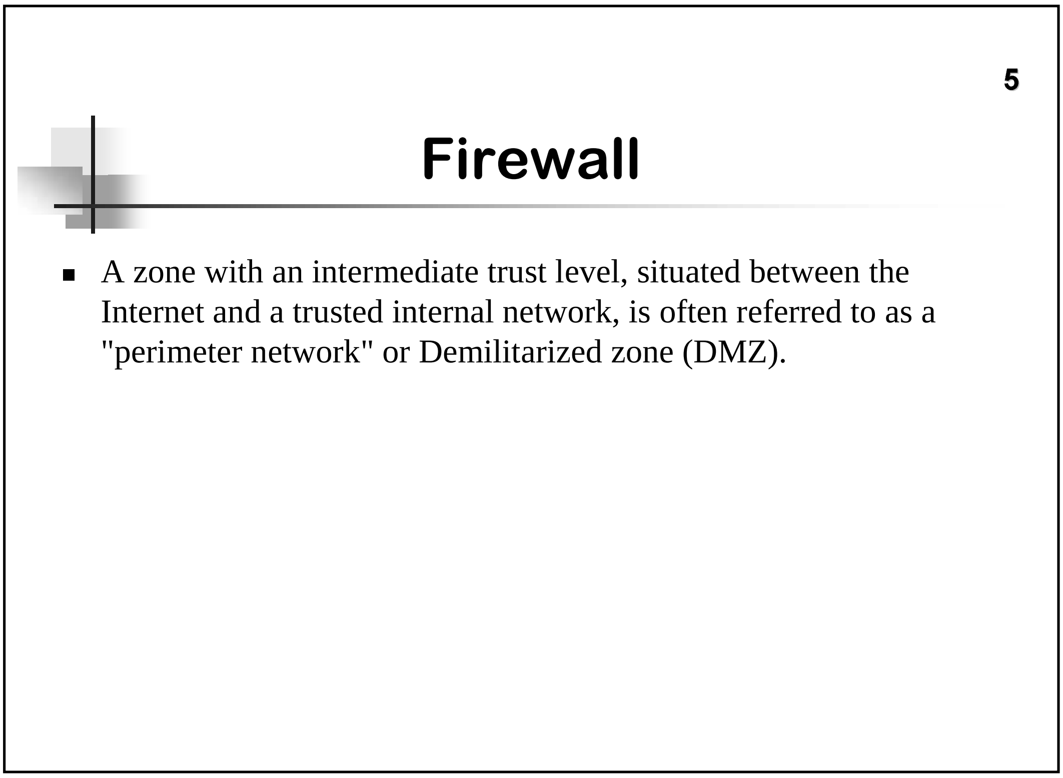
Firewall in construction

- A passive fire protection item that is required to have a special and unique Fire-resistance rating and is subject to stringent bounding.
- It is a fire-resistance rated wall assembly that has been constructed in such a manner as to subdivide a building into legal and practical segments in accordance with the locally applicable building code.



Firewall in IT

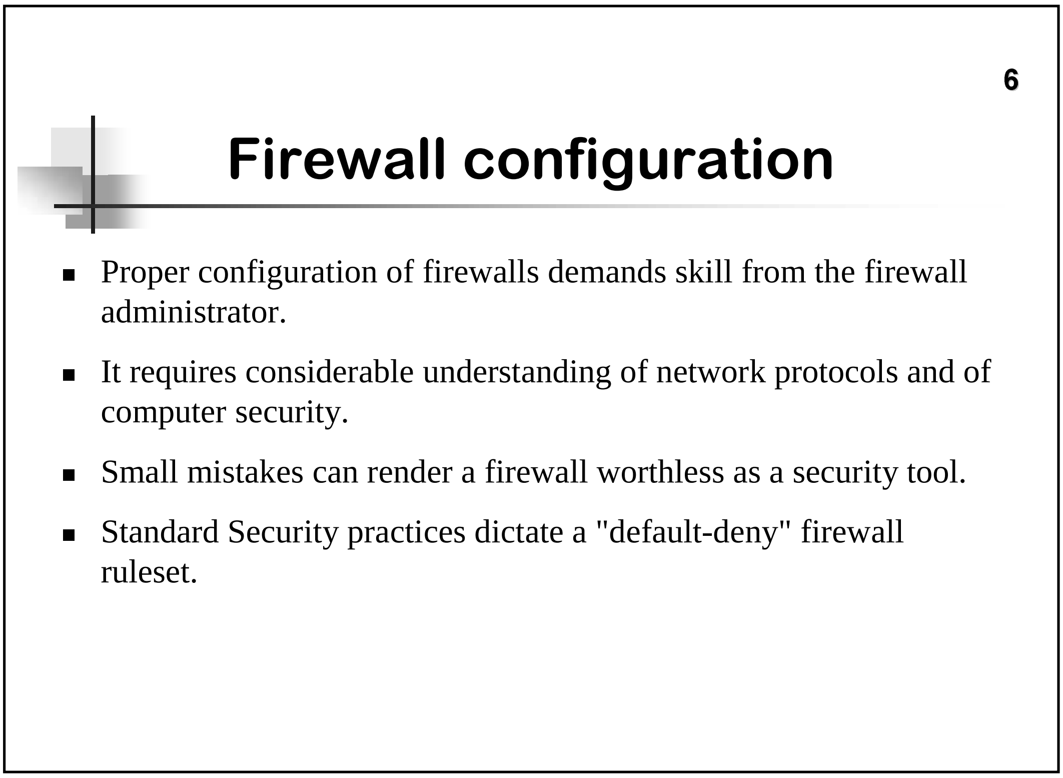
- A security device / software which is configured to permit, deny or proxy data connections set and configured by the organization's security policy.
- A firewall's basic task is to control traffic between computer networks with different zones of trust:
  - Zone with no trust (Internet)
  - with high trust (internal network)
- The ultimate goal is to provide controlled interfaces between zones of differing trust levels through the enforcement of a security policy and connectivity model.



5

Firewall

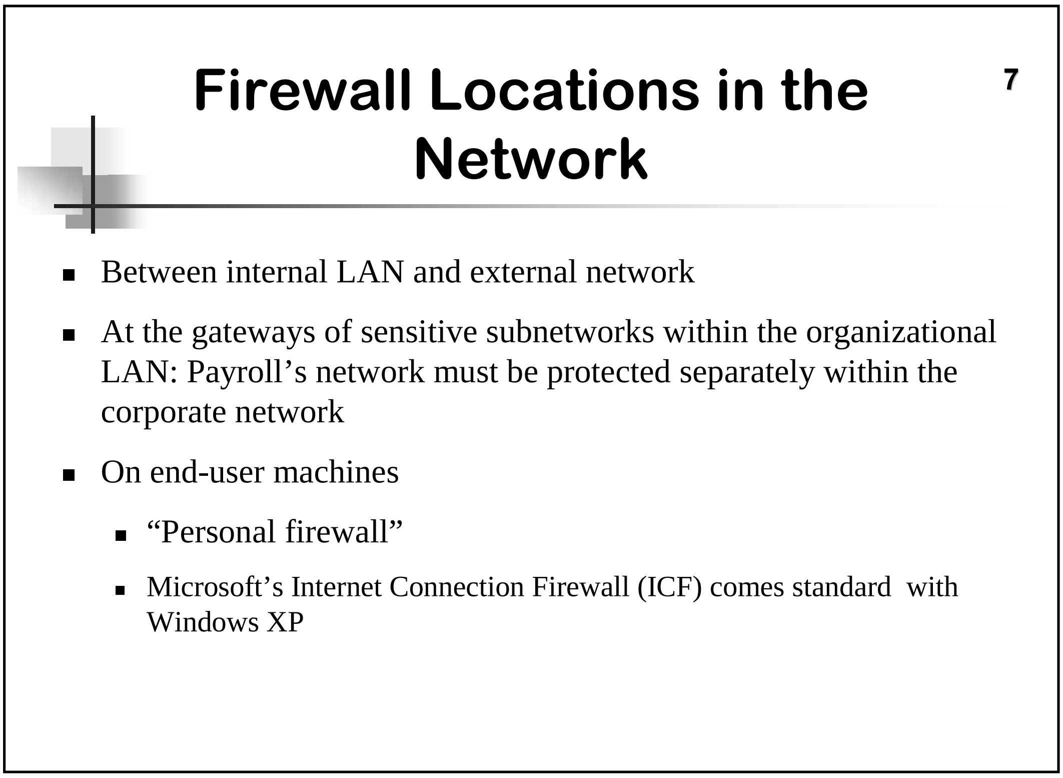
- A zone with an intermediate trust level, situated between the Internet and a trusted internal network, is often referred to as a "perimeter network" or Demilitarized zone (DMZ).



6

Firewall configuration

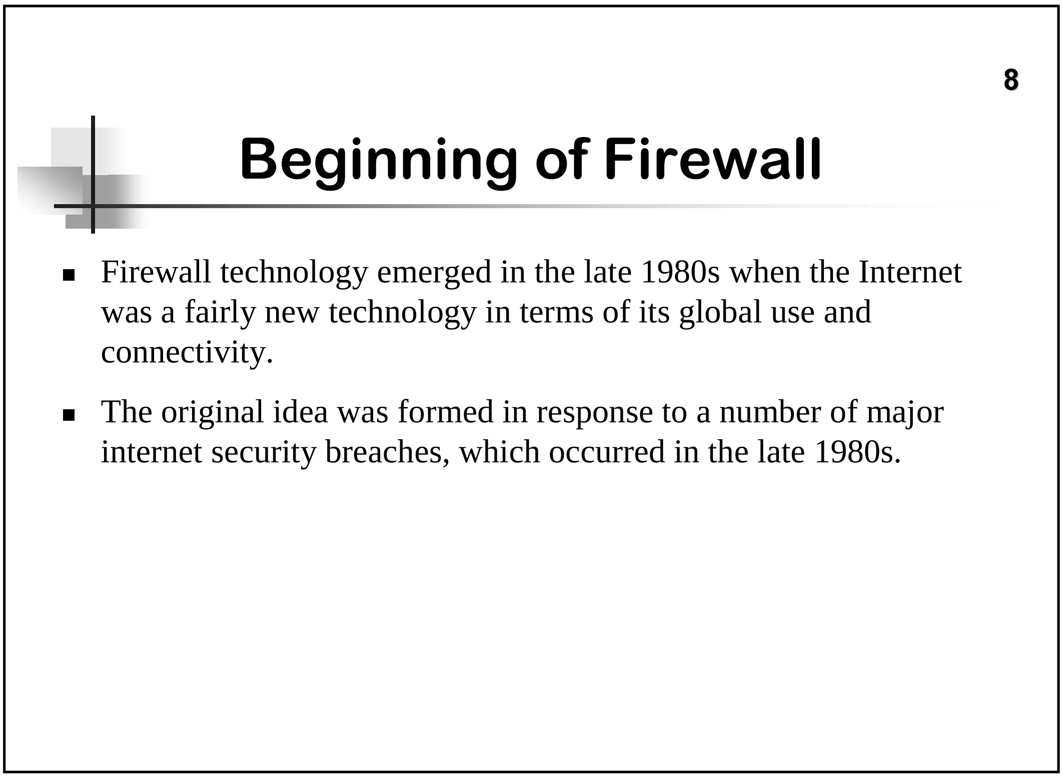
- Proper configuration of firewalls demands skill from the firewall administrator.
- It requires considerable understanding of network protocols and of computer security.
- Small mistakes can render a firewall worthless as a security tool.
- Standard Security practices dictate a "default-deny" firewall ruleset.



7

Firewall Locations in the Network

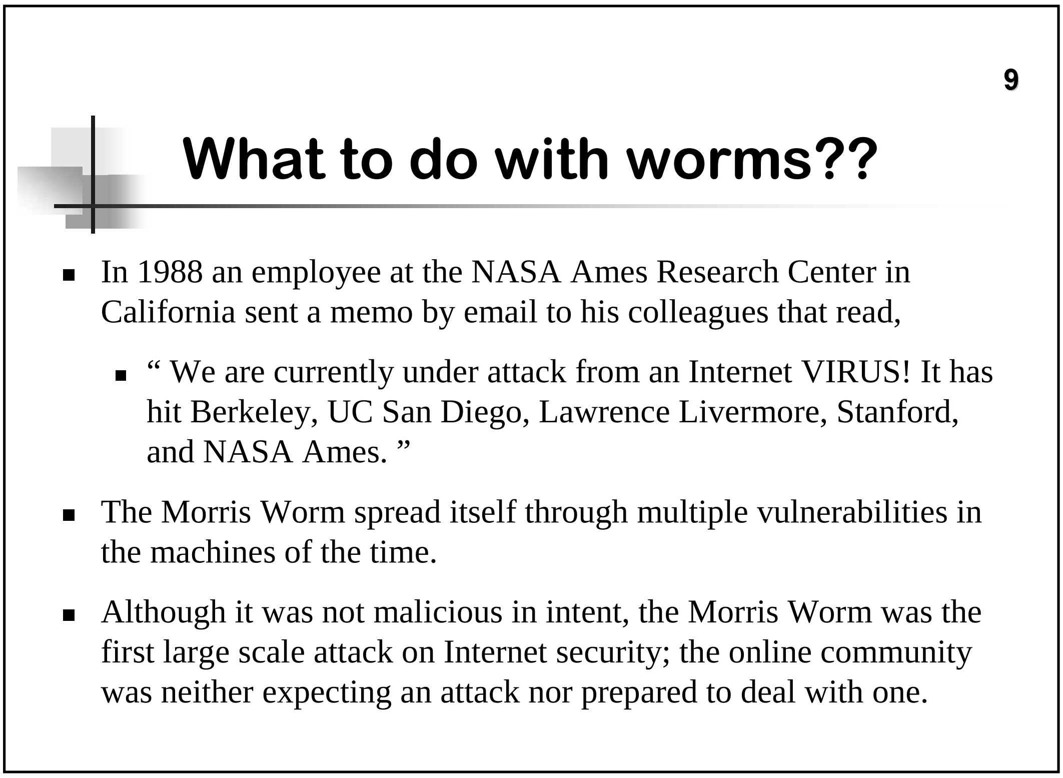
- Between internal LAN and external network
- At the gateways of sensitive subnetworks within the organizational LAN: Payroll’s network must be protected separately within the corporate network
- On end-user machines
    - “Personal firewall”
    - Microsoft’s Internet Connection Firewall (ICF) comes standard with Windows XP



8

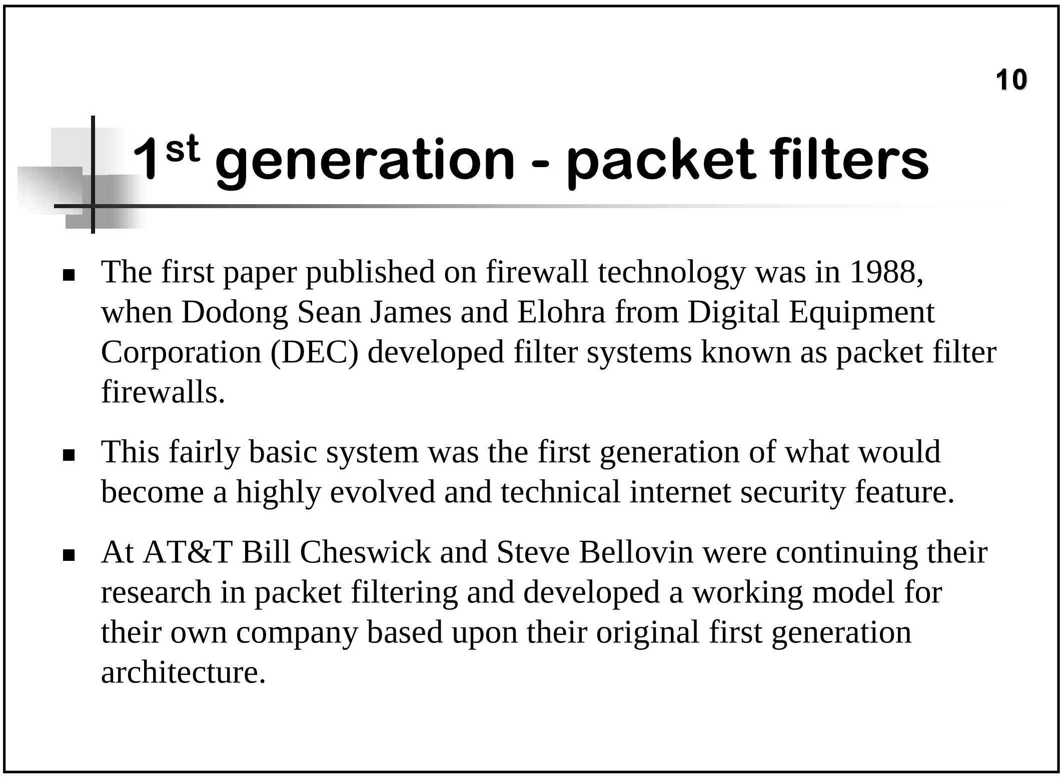
Beginning of Firewall

- Firewall technology emerged in the late 1980s when the Internet was a fairly new technology in terms of its global use and connectivity.
- The original idea was formed in response to a number of major internet security breaches, which occurred in the late 1980s.



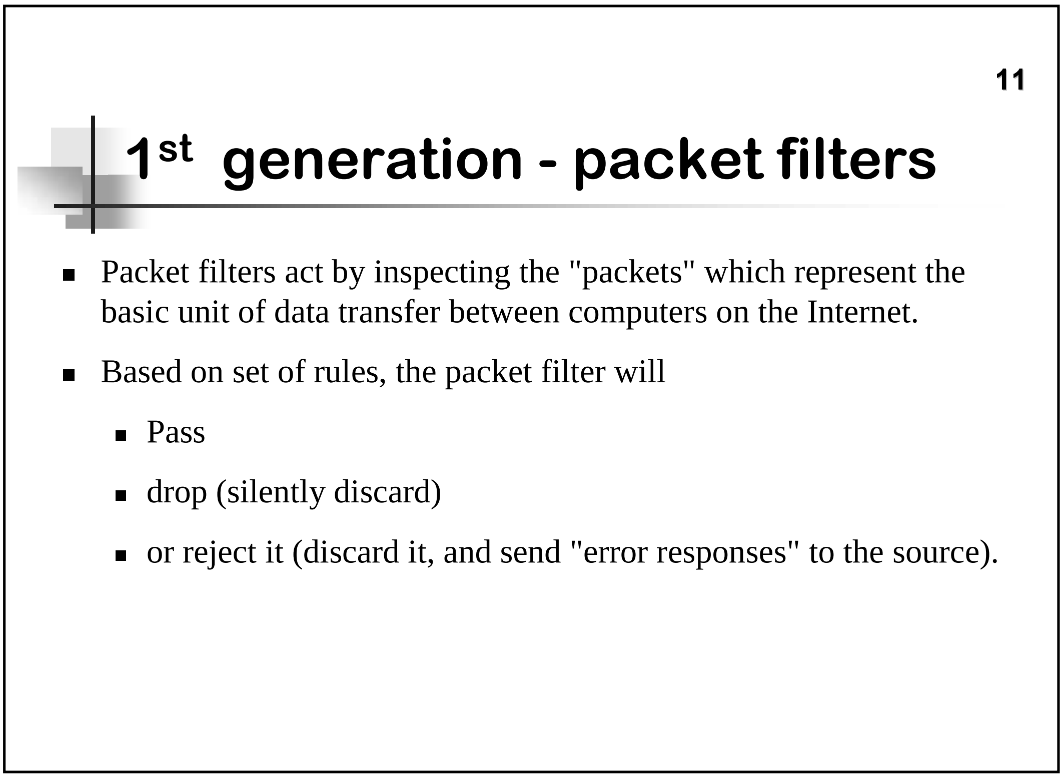
What to do with worms??

- In 1988 an employee at the NASA Ames Research Center in California sent a memo by email to his colleagues that read,
  - “ We are currently under attack from an Internet VIRUS! It has hit Berkeley, UC San Diego, Lawrence Livermore, Stanford, and NASA Ames. ”
- The Morris Worm spread itself through multiple vulnerabilities in the machines of the time.
- Although it was not malicious in intent, the Morris Worm was the first large scale attack on Internet security; the online community was neither expecting an attack nor prepared to deal with one.



1st generation - packet filters

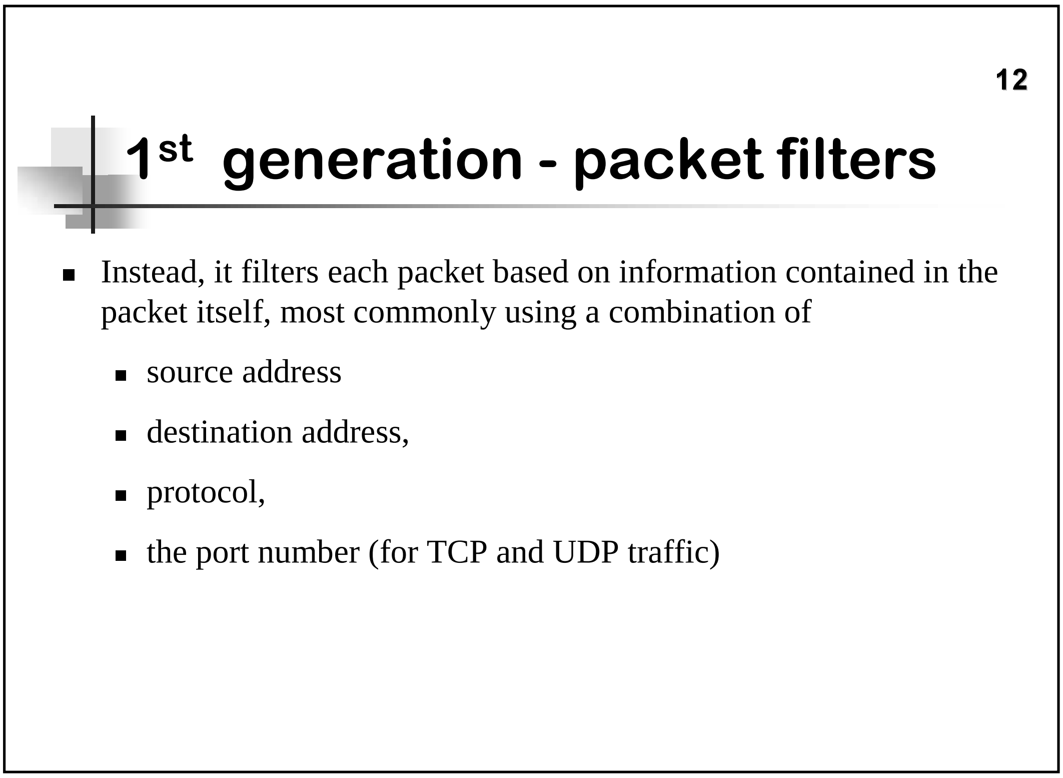
- The first paper published on firewall technology was in 1988, when Dodong Sean James and Elohra from Digital Equipment Corporation (DEC) developed filter systems known as packet filter firewalls.
- This fairly basic system was the first generation of what would become a highly evolved and technical internet security feature.
- At AT&T Bill Cheswick and Steve Bellovin were continuing their research in packet filtering and developed a working model for their own company based upon their original first generation architecture.



11

1st generation - packet filters

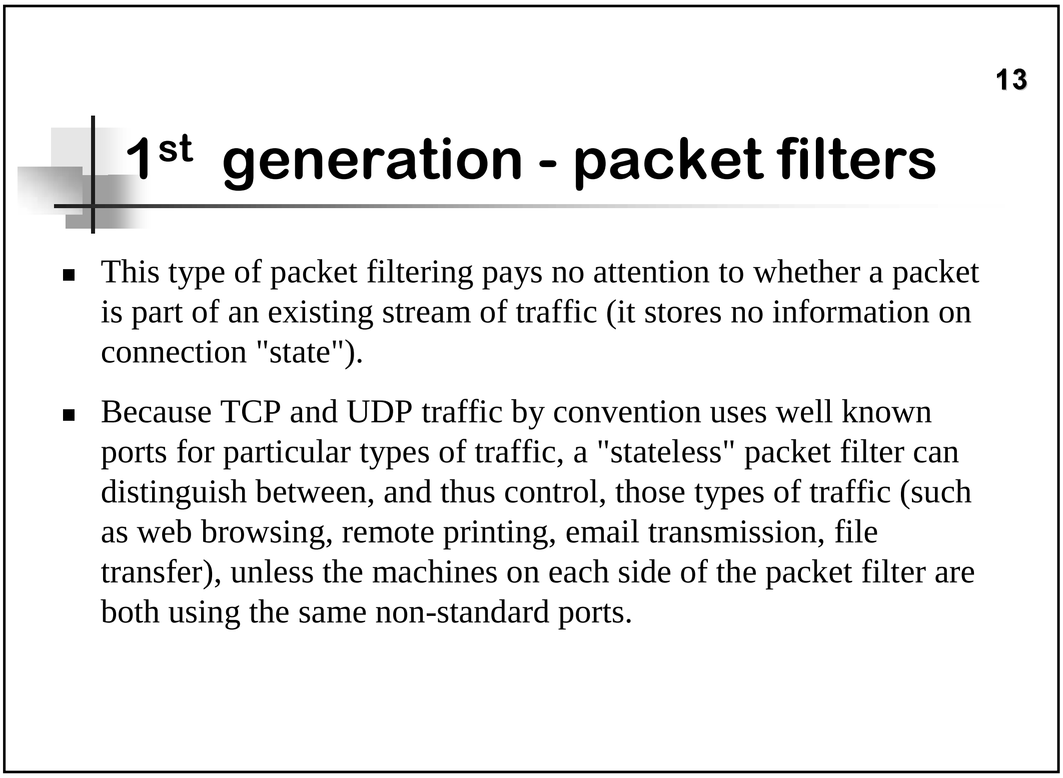
- Packet filters act by inspecting the "packets" which represent the basic unit of data transfer between computers on the Internet.
- Based on set of rules, the packet filter will
  - Pass
  - drop (silently discard)
  - or reject it (discard it, and send "error responses" to the source).



12

1st generation - packet filters

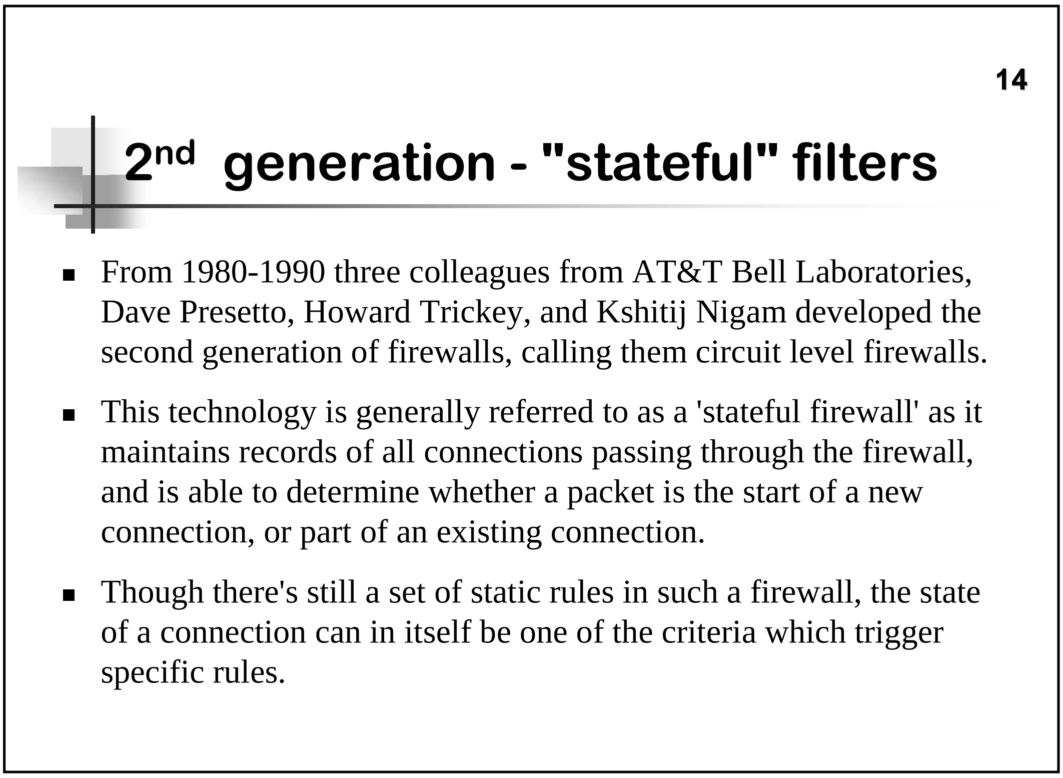
- Instead, it filters each packet based on information contained in the packet itself, most commonly using a combination of
  - source address
  - destination address,
  - protocol,
  - the port number (for TCP and UDP traffic)



13

1st generation - packet filters

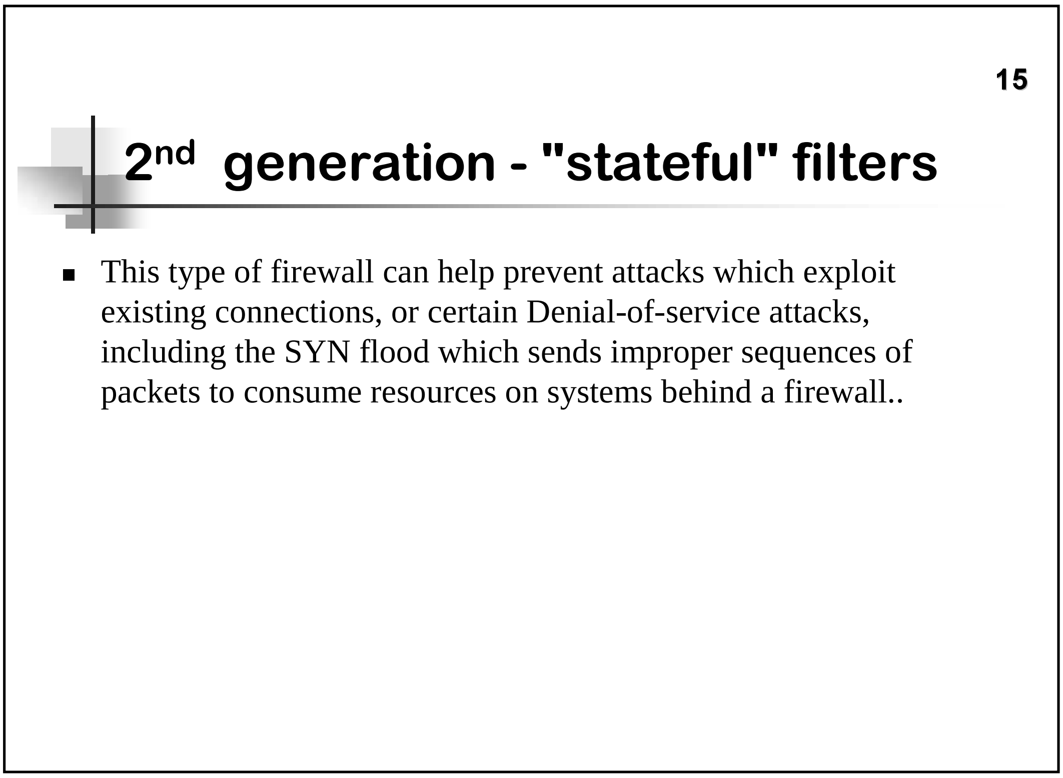
- This type of packet filtering pays no attention to whether a packet is part of an existing stream of traffic (it stores no information on connection "state").
- Because TCP and UDP traffic by convention uses well known ports for particular types of traffic, a "stateless" packet filter can distinguish between, and thus control, those types of traffic (such as web browsing, remote printing, email transmission, file transfer), unless the machines on each side of the packet filter are both using the same non-standard ports.



14

2nd generation - "stateful" filters

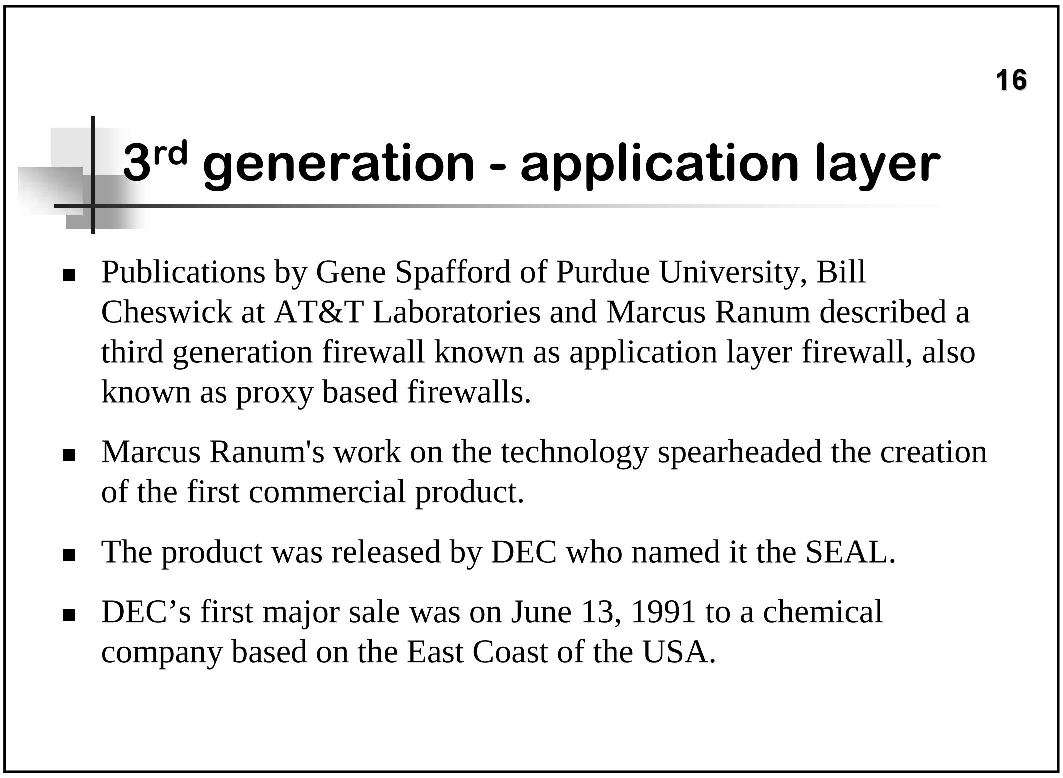
- From 1980-1990 three colleagues from AT&T Bell Laboratories, Dave Presetto, Howard Trickey, and Kshitij Nigam developed the second generation of firewalls, calling them circuit level firewalls.
- This technology is generally referred to as a 'stateful firewall' as it maintains records of all connections passing through the firewall, and is able to determine whether a packet is the start of a new connection, or part of an existing connection.
- Though there's still a set of static rules in such a firewall, the state of a connection can in itself be one of the criteria which trigger specific rules.



15

2nd generation - "stateful" filters

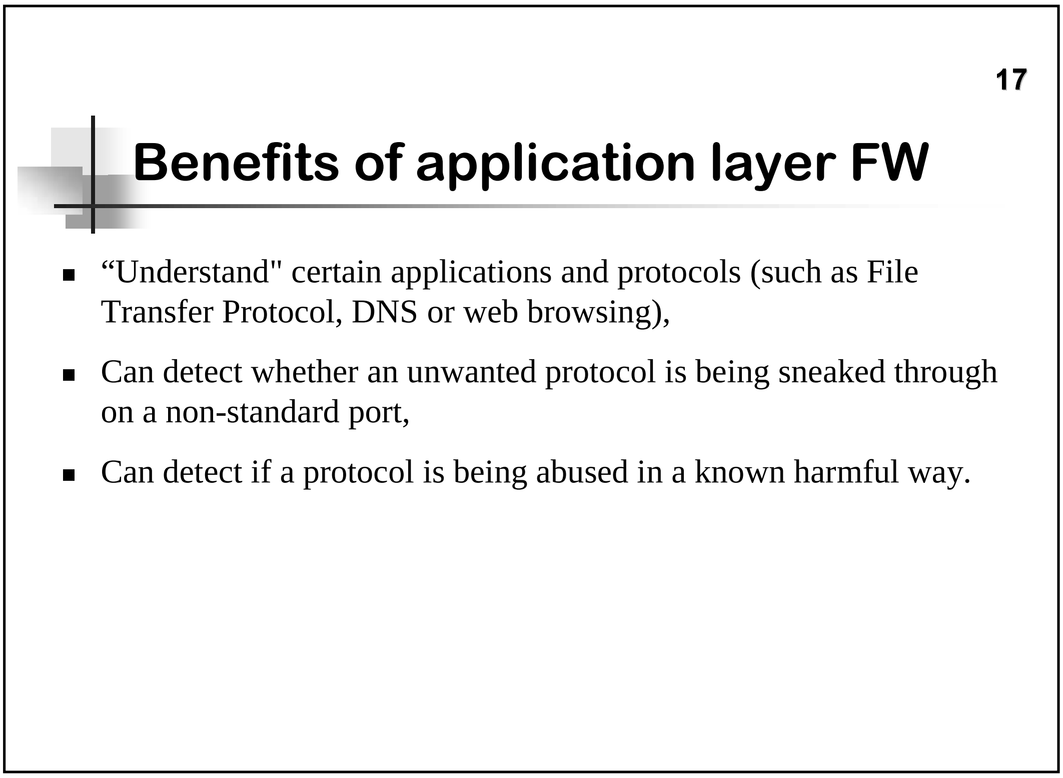
- This type of firewall can help prevent attacks which exploit existing connections, or certain Denial-of-service attacks, including the SYN flood which sends improper sequences of packets to consume resources on systems behind a firewall..



16

3rd generation - application layer

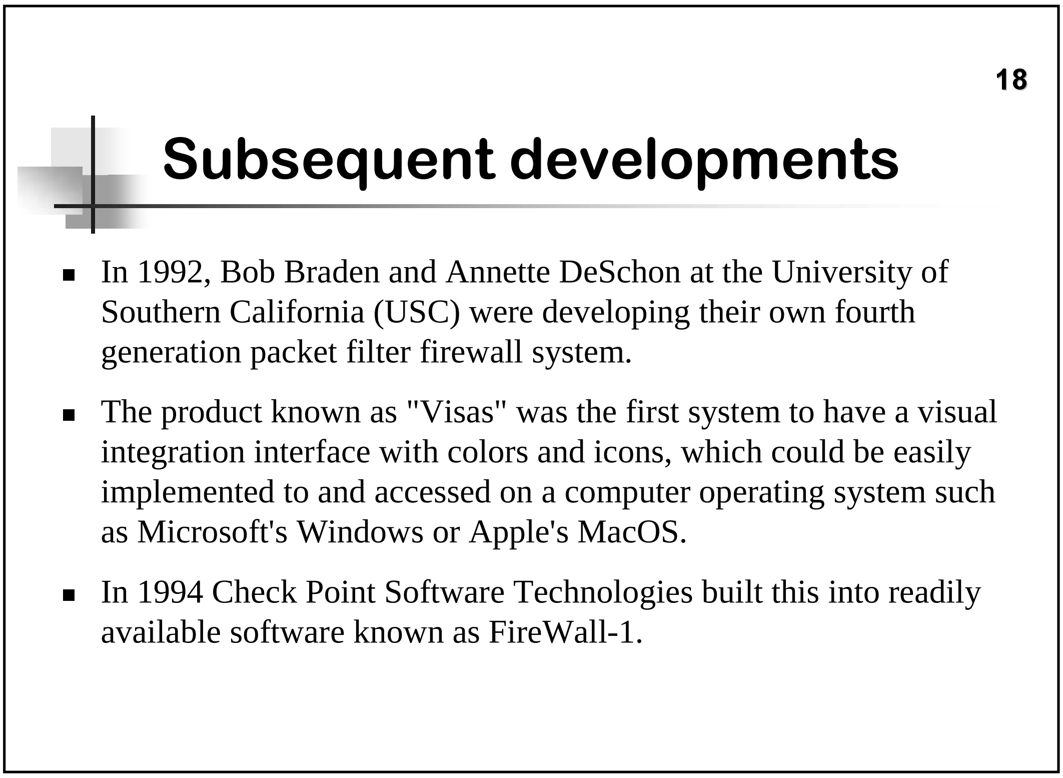
- Publications by Gene Spafford of Purdue University, Bill Cheswick at AT&T Laboratories and Marcus Ranum described a third generation firewall known as application layer firewall, also known as proxy based firewalls.
- Marcus Ranum's work on the technology spearheaded the creation of the first commercial product.
- The product was released by DEC who named it the SEAL.
- DEC’s first major sale was on June 13, 1991 to a chemical company based on the East Coast of the USA.



17

Benefits of application layer FW

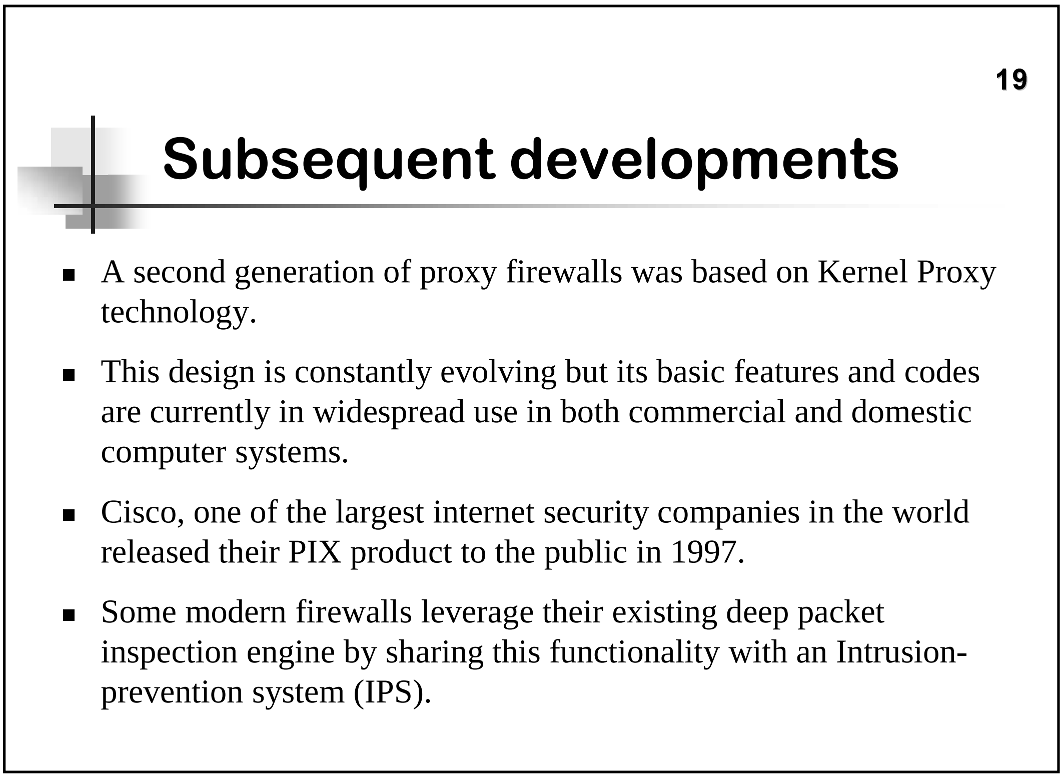
- “Understand" certain applications and protocols (such as File Transfer Protocol, DNS or web browsing),
- Can detect whether an unwanted protocol is being sneaked through on a non-standard port,
- Can detect if a protocol is being abused in a known harmful way.



18

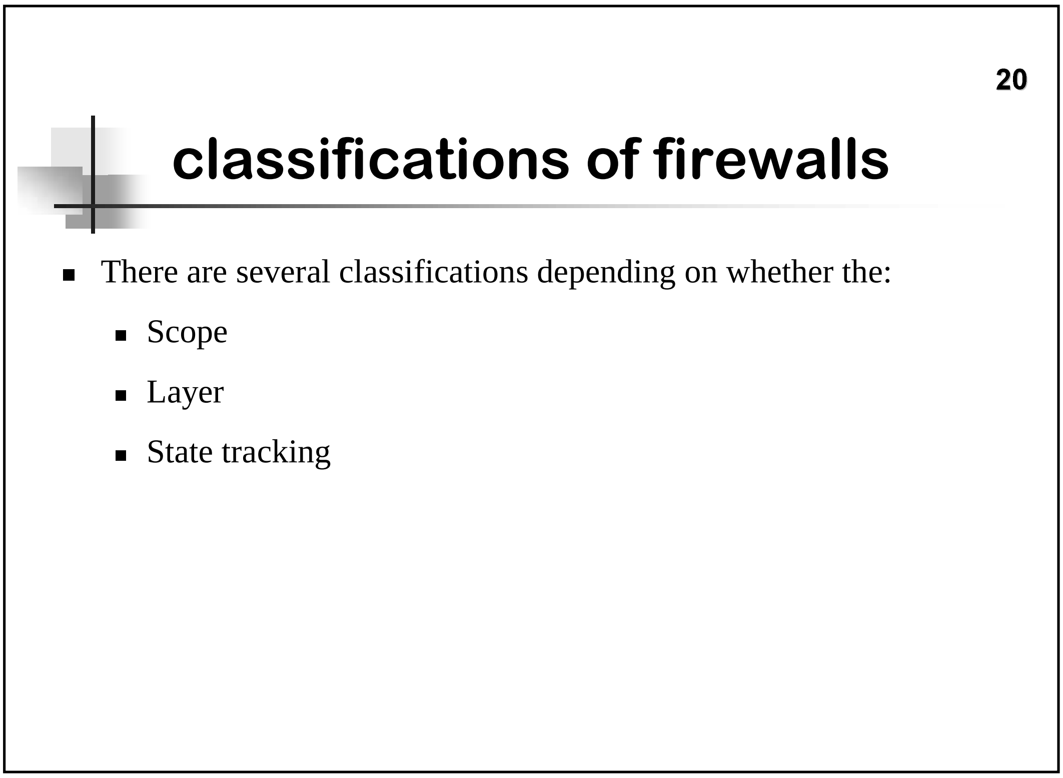
Subsequent developments

- In 1992, Bob Braden and Annette DeSchon at the University of Southern California (USC) were developing their own fourth generation packet filter firewall system.
- The product known as "Visas" was the first system to have a visual integration interface with colors and icons, which could be easily implemented to and accessed on a computer operating system such as Microsoft's Windows or Apple's MacOS.
- In 1994 Check Point Software Technologies built this into readily available software known as FireWall-1.



Subsequent developments

- A second generation of proxy firewalls was based on Kernel Proxy technology.
- This design is constantly evolving but its basic features and codes are currently in widespread use in both commercial and domestic computer systems.
- Cisco, one of the largest internet security companies in the world released their PIX product to the public in 1997.
- Some modern firewalls leverage their existing deep packet inspection engine by sharing this functionality with an Intrusion-prevention system (IPS).



classifications of firewalls

- There are several classifications depending on whether the:
  - Scope
  - Layer
  - State tracking

21

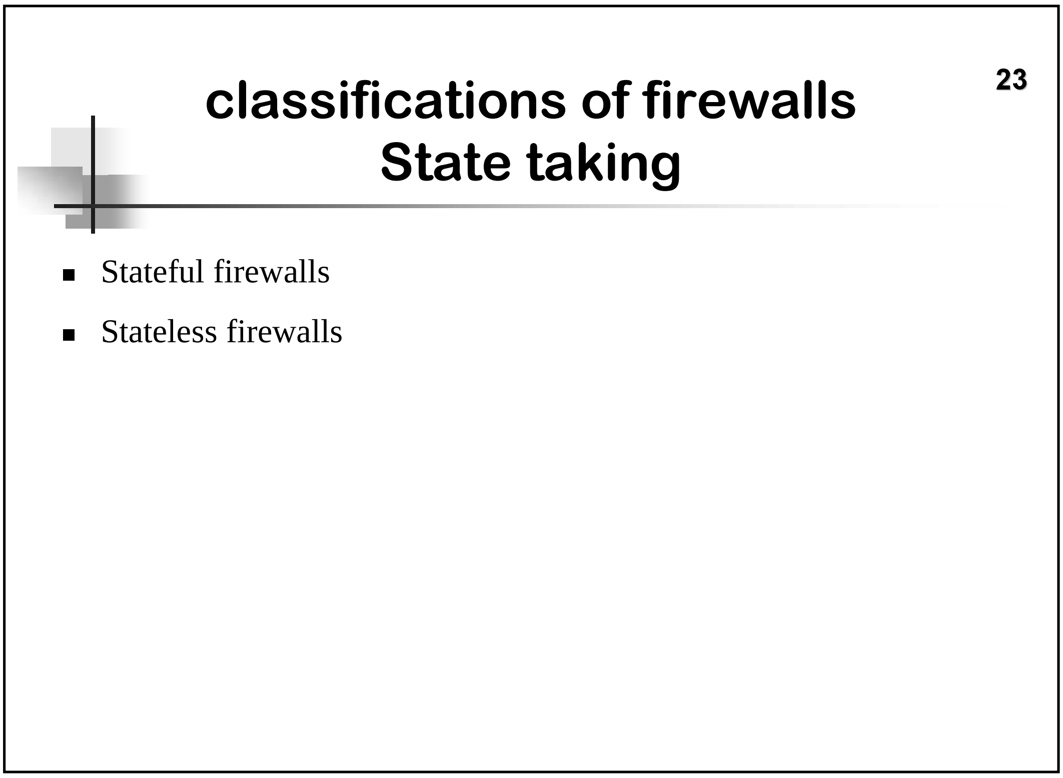
classifications of firewalls Scope

- Personal firewalls, a software application which normally filters traffic entering or leaving a single computer. This filtering may be based on the traffic itself or on the identity of the process which is attempting to listen for or send data.
- Network firewalls, normally running on a dedicated network device or computer positioned on the boundary of two or more networks or DMZs (demilitarized zones). Such a firewall filters all traffic entering or leaving the connected networks.

22

classifications of firewalls Layer

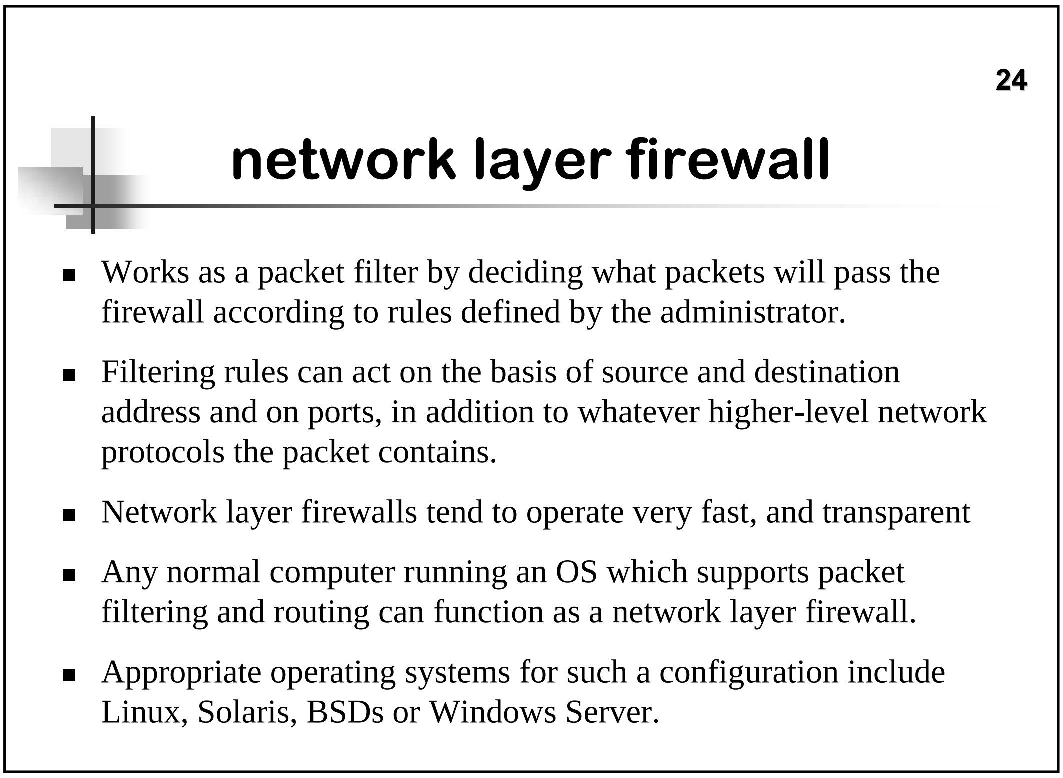
- Network layer firewalls (iptables)
- Application layer firewalls (TCP Wrappers)
- Application firewalls (ftpaccess file)
- Network-layer and Application-layer firewalls may overlap, even though the personal firewall does not serve a network; indeed, single systems have implemented both together.



23

classifications of firewalls State taking

- Stateful firewalls
- Stateless firewalls



24

network layer firewall

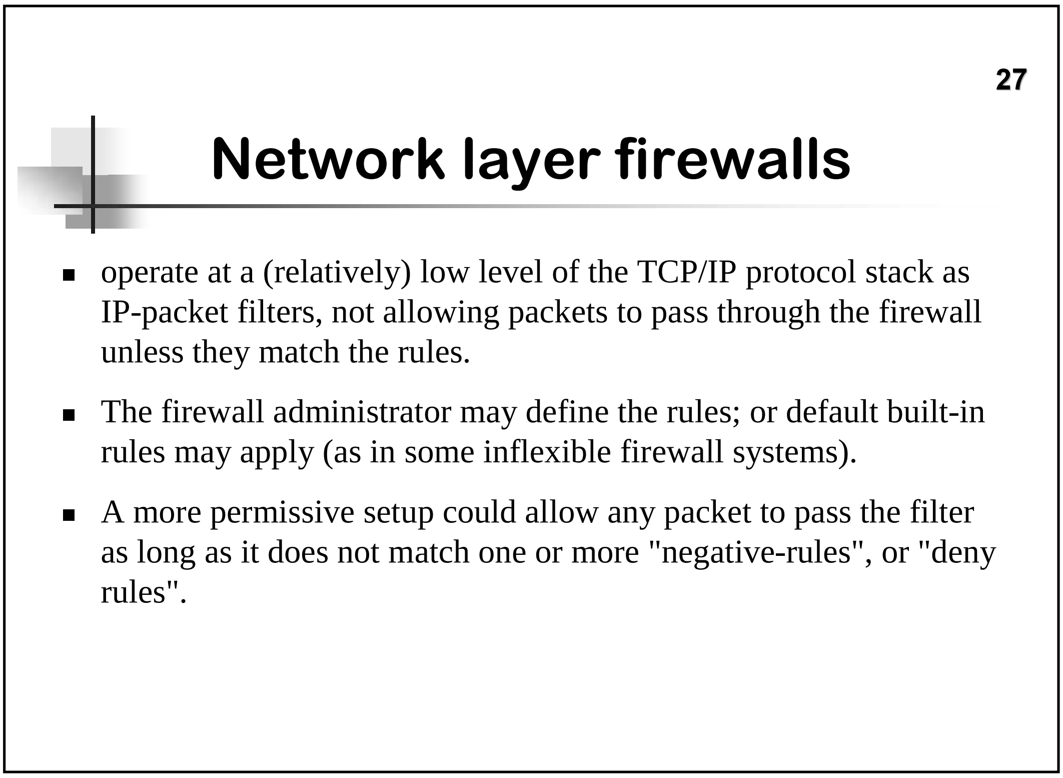
- Works as a packet filter by deciding what packets will pass the firewall according to rules defined by the administrator.
- Filtering rules can act on the basis of source and destination address and on ports, in addition to whatever higher-level network protocols the packet contains.
- Network layer firewalls tend to operate very fast, and transparent
- Any normal computer running an OS which supports packet filtering and routing can function as a network layer firewall.
- Appropriate operating systems for such a configuration include Linux, Solaris, BSDs or Windows Server.

Stateful firewall

- Firewalls hold some information on the state of connections (for example: established or not, initiation, handshaking, data or breaking down the connection) as part of their rules (e.g. only hosts inside the firewall can establish connections on a certain port).

Stateful firewall

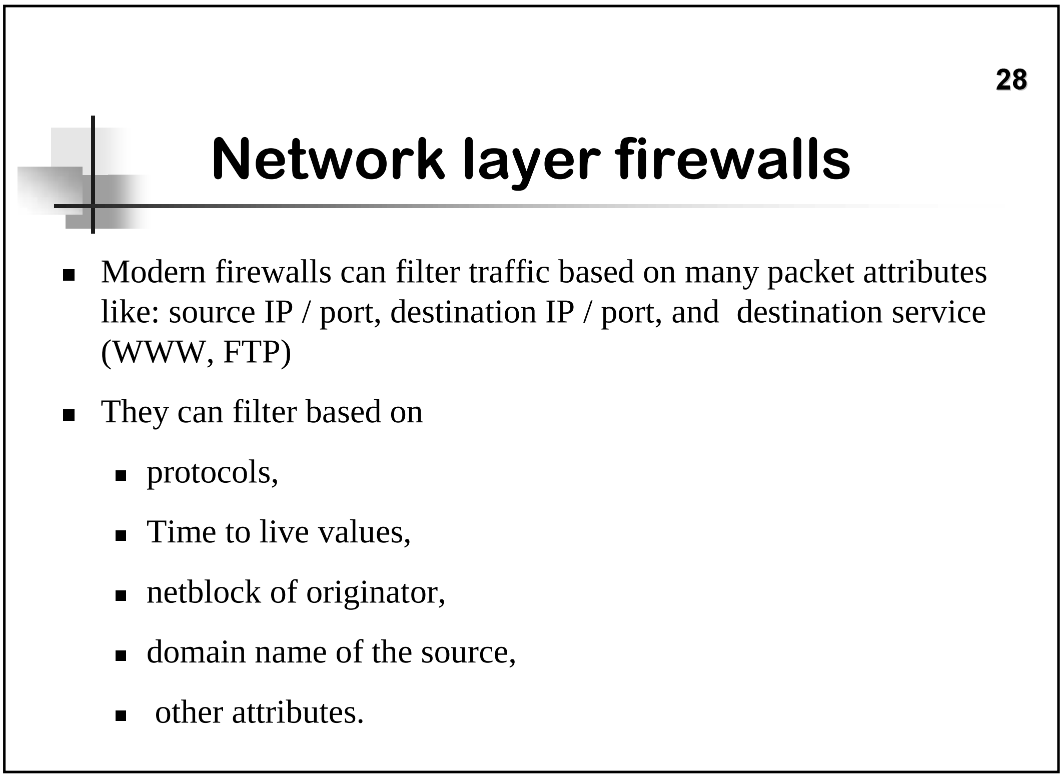
- Stateless firewalls have packet-filtering capabilities but cannot make more complex decisions on what stage communications between hosts have reached. Stateless firewalls therefore offer less security. Stateless firewalls somewhat resemble a router in their ability to filter packets.



27

Network layer firewalls

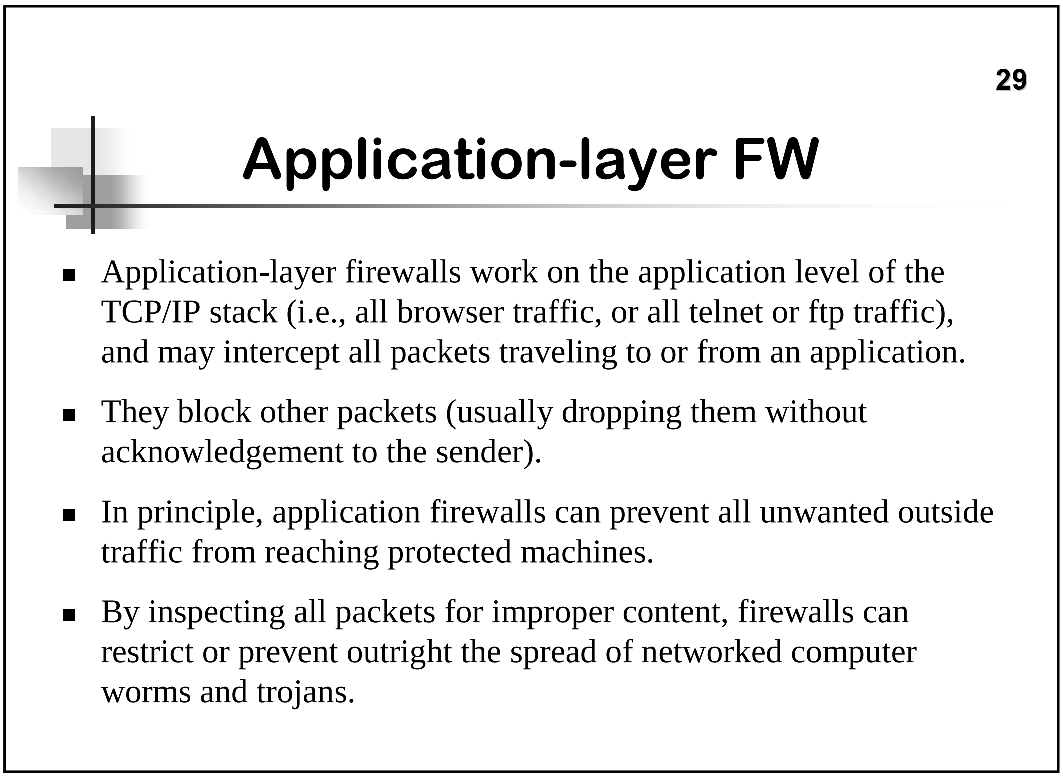
- operate at a (relatively) low level of the TCP/IP protocol stack as IP-packet filters, not allowing packets to pass through the firewall unless they match the rules.
- The firewall administrator may define the rules; or default built-in rules may apply (as in some inflexible firewall systems).
- A more permissive setup could allow any packet to pass the filter as long as it does not match one or more "negative-rules", or "deny rules".



28

Network layer firewalls

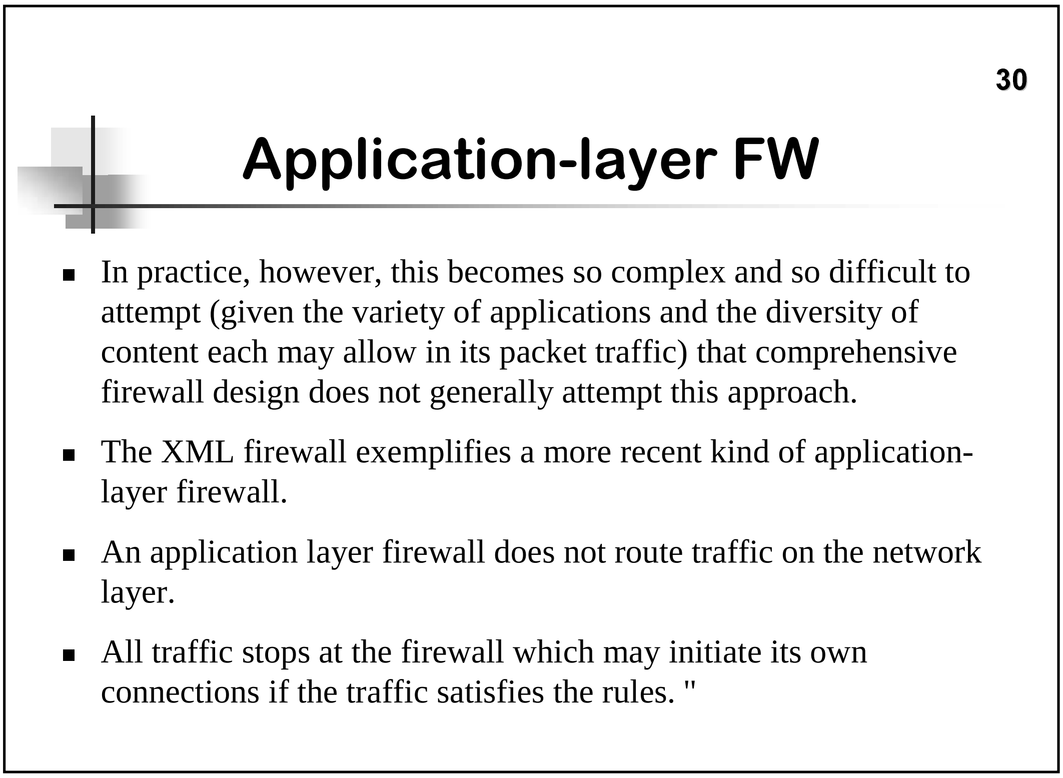
- Modern firewalls can filter traffic based on many packet attributes like: source IP / port, destination IP / port, and destination service (WWW, FTP)
- They can filter based on
  - protocols,
  - Time to live values,
  - netblock of originator,
  - domain name of the source,
  - other attributes.



29

Application-layer FW

- Application-layer firewalls work on the application level of the TCP/IP stack (i.e., all browser traffic, or all telnet or ftp traffic), and may intercept all packets traveling to or from an application.
- They block other packets (usually dropping them without acknowledgement to the sender).
- In principle, application firewalls can prevent all unwanted outside traffic from reaching protected machines.
- By inspecting all packets for improper content, firewalls can restrict or prevent outright the spread of networked computer worms and trojans.



30

Application-layer FW

- In practice, however, this becomes so complex and so difficult to attempt (given the variety of applications and the diversity of content each may allow in its packet traffic) that comprehensive firewall design does not generally attempt this approach.
- The XML firewall exemplifies a more recent kind of application-layer firewall.
- An application layer firewall does not route traffic on the network layer.
- All traffic stops at the firewall which may initiate its own connections if the traffic satisfies the rules. "

31

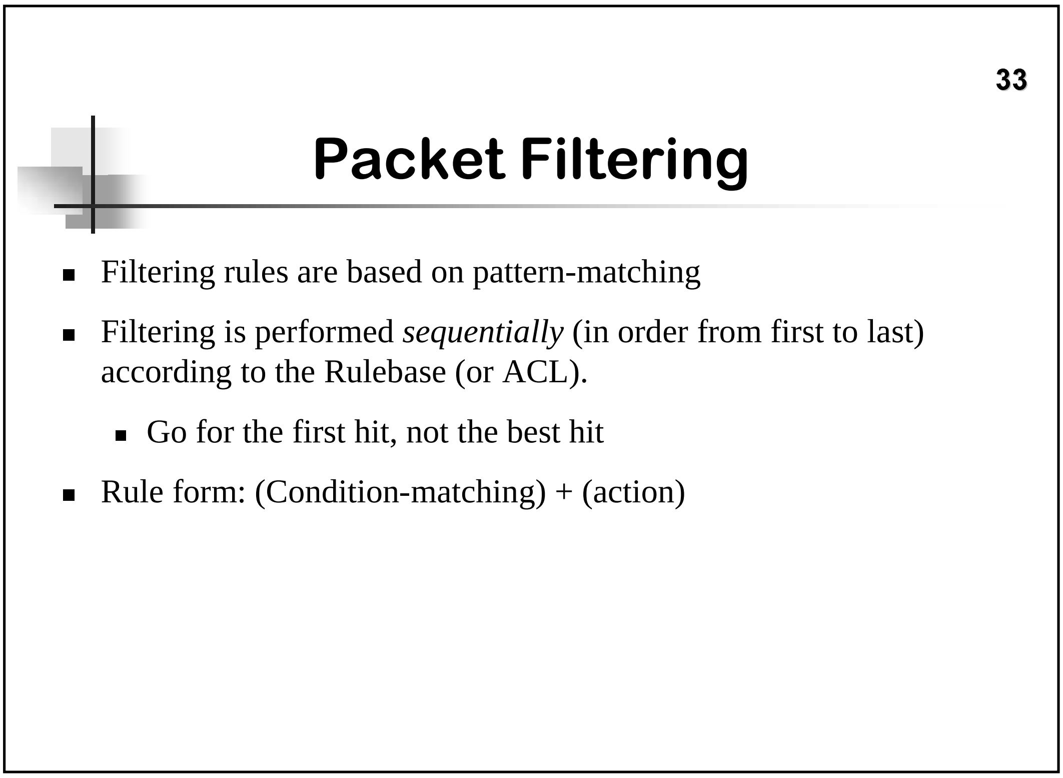
Network address translation

- Firewalls often have network address translation (NAT) functionality, and the hosts protected behind a firewall commonly have addresses in the "private address range", as defined in RFC 1918.
- Firewalls often have such functionality to hide the true address of protected hosts.

32

Packet Filtering

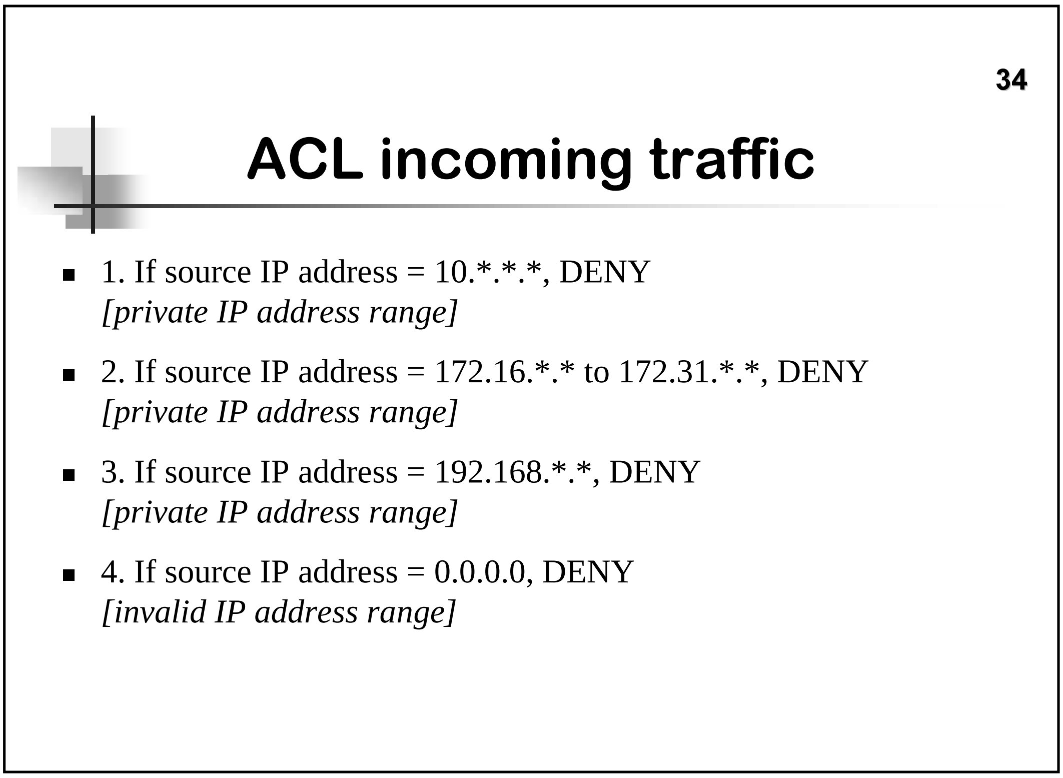
- For each packet, firewall decides whether to allow it to proceed
  - Decision must be made on per-packet basis
    - Stateless; cannot examine packet’s context (TCP connection, application to which it belongs, etc.)
- To decide, use information available in the packet
  - IP source and destination addresses, ports
  - Protocol identifier (TCP, UDP, ICMP, etc.)
  - TCP flags (SYN, ACK, RST, PSH, FIN)
  - ICMP message type



33

Packet Filtering

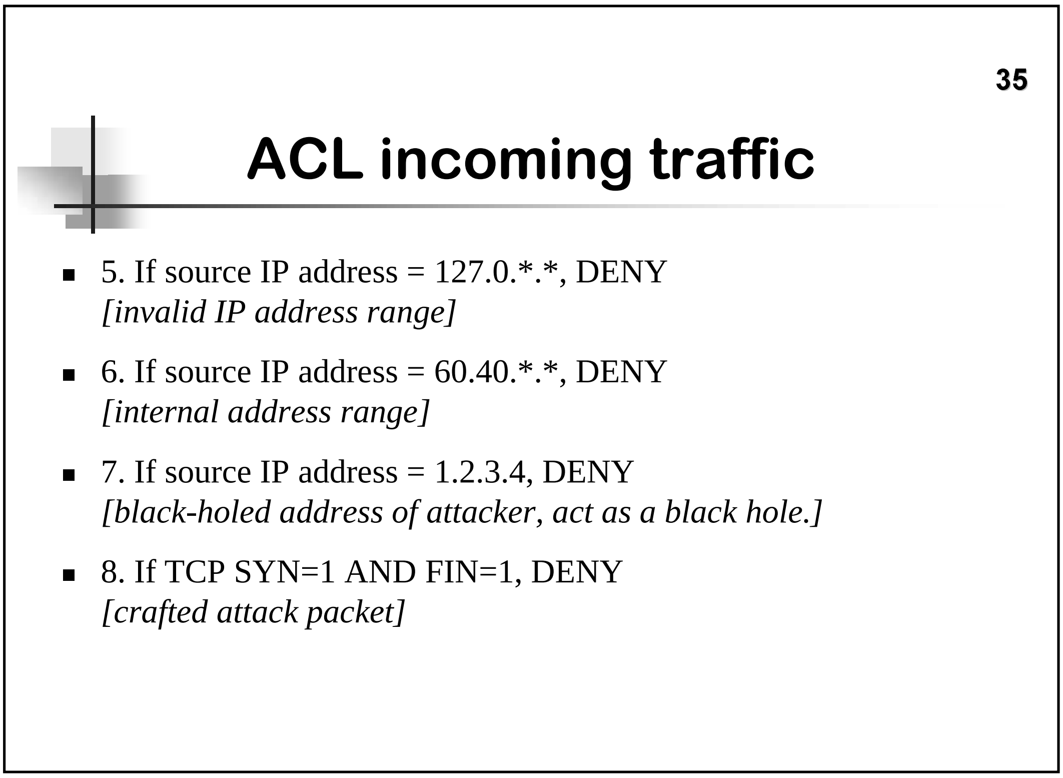
- Filtering rules are based on pattern-matching
- Filtering is performed *sequentially* (in order from first to last) according to the Rulebase (or ACL).
  - Go for the first hit, not the best hit
- Rule form: (Condition-matching) + (action)



34

ACL incoming traffic

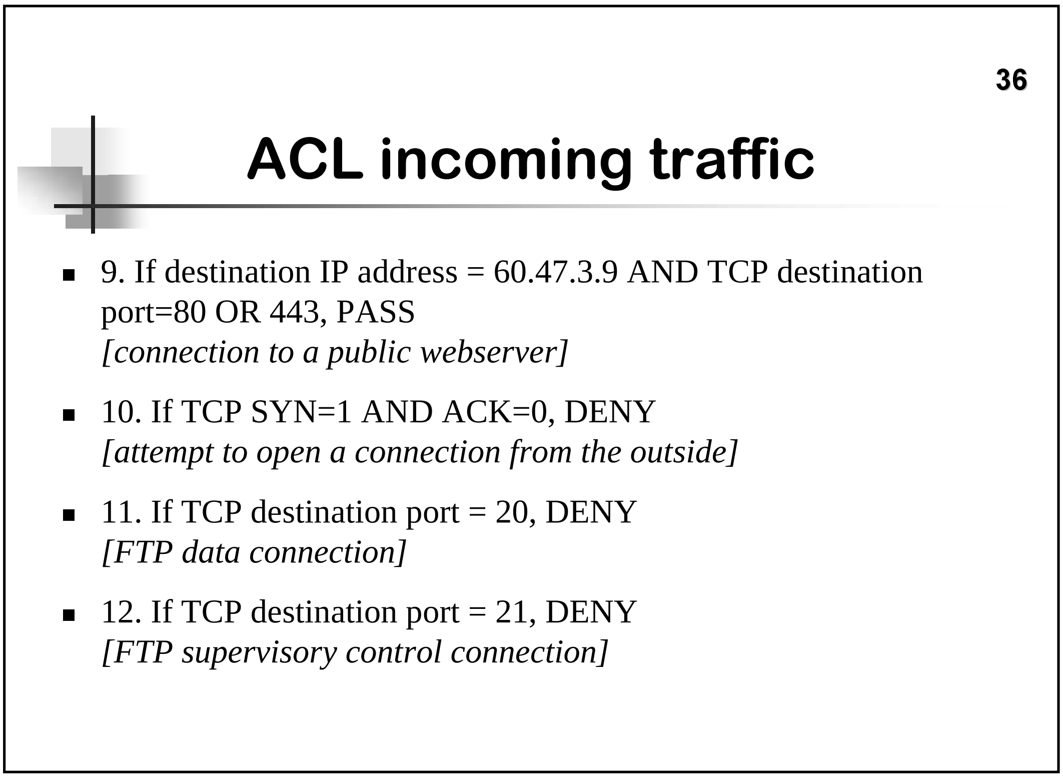
- 1. If source IP address = 10.*.*.*, DENY
  [private IP address range]
- 2. If source IP address = 172.16.*.* to 172.31.*.*, DENY
  [private IP address range]
- 3. If source IP address = 192.168.*.*, DENY
  [private IP address range]
- 4. If source IP address = 0.0.0.0, DENY
  [invalid IP address range]



35

ACL incoming traffic

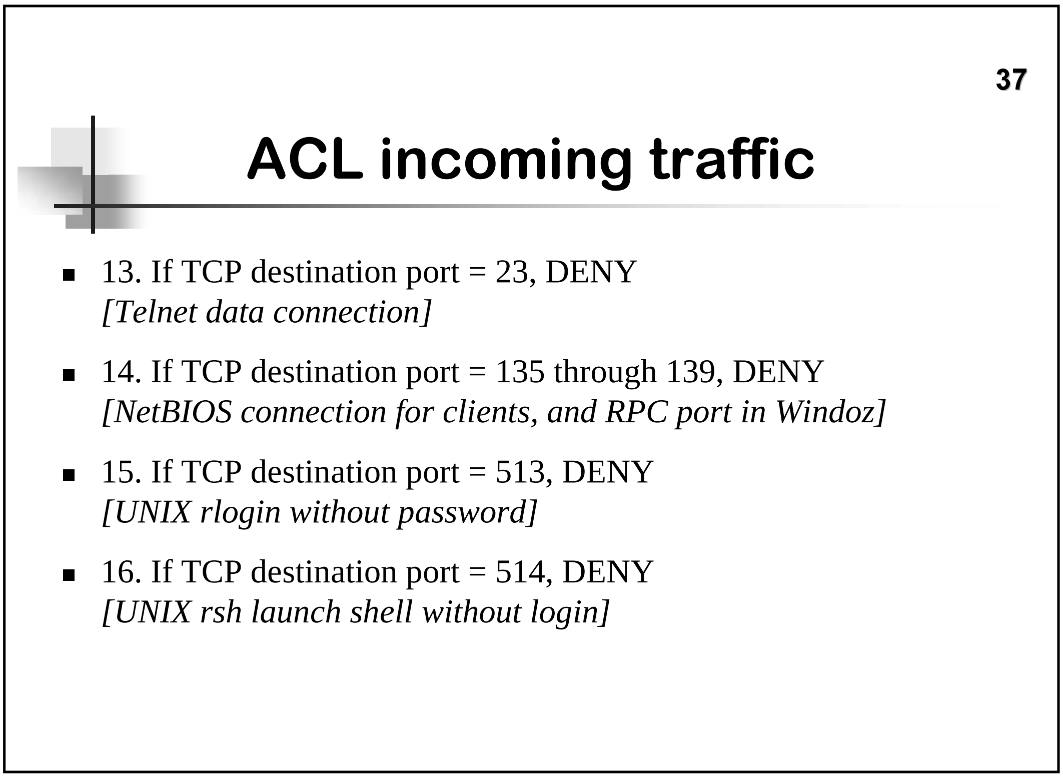
- 5. If source IP address = 127.0.*.*, DENY
  [invalid IP address range]
- 6. If source IP address = 60.40.*.*, DENY
  [internal address range]
- 7. If source IP address = 1.2.3.4, DENY
  [black-holed address of attacker, act as a black hole.]
- 8. If TCP SYN=1 AND FIN=1, DENY
  [crafted attack packet]



36

ACL incoming traffic

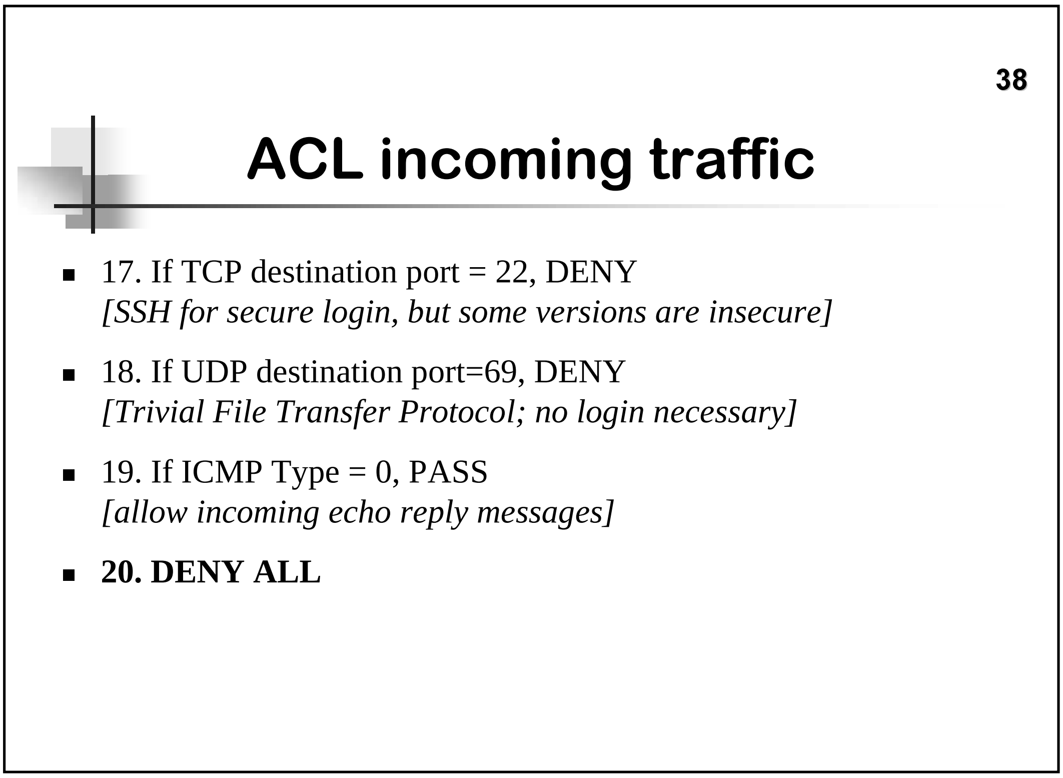
- 9. If destination IP address = 60.47.3.9 AND TCP destination port=80 OR 443, PASS
  [connection to a public webserver]
- 10. If TCP SYN=1 AND ACK=0, DENY
  [attempt to open a connection from the outside]
- 11. If TCP destination port = 20, DENY
  [FTP data connection]
- 12. If TCP destination port = 21, DENY
  [FTP supervisory control connection]



37

ACL incoming traffic

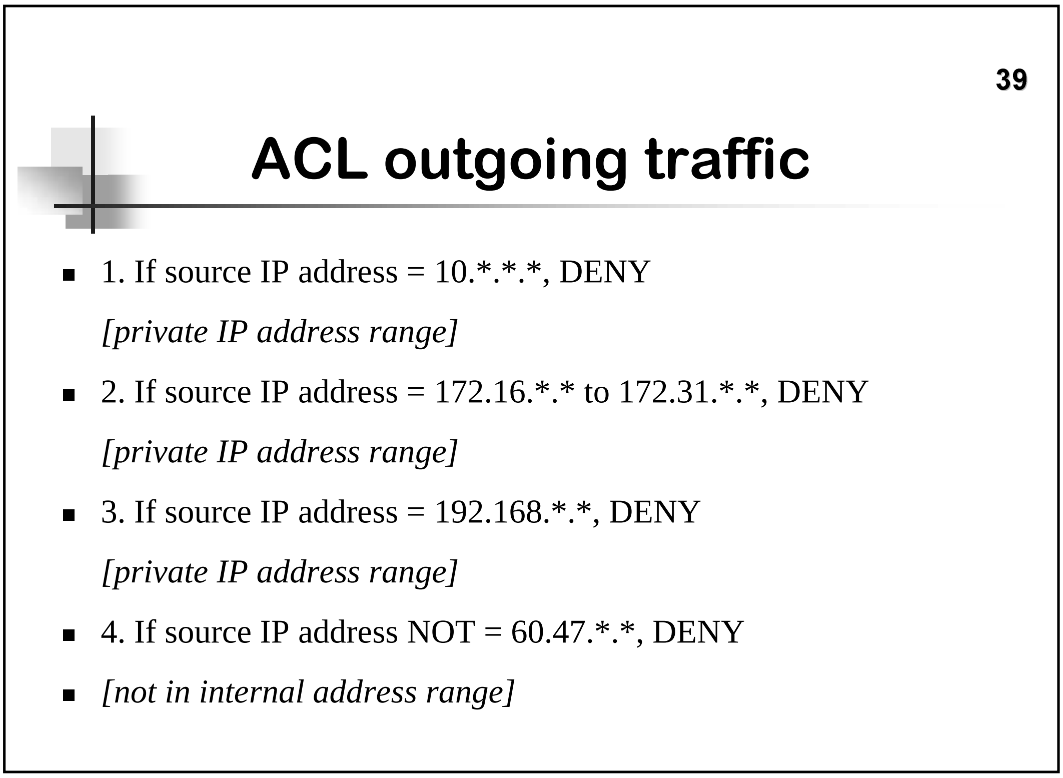
- 13. If TCP destination port = 23, DENY
  [Telnet data connection]
- 14. If TCP destination port = 135 through 139, DENY
  [NetBIOS connection for clients, and RPC port in Windoz]
- 15. If TCP destination port = 513, DENY
  [UNIX rlogin without password]
- 16. If TCP destination port = 514, DENY
  [UNIX rsh launch shell without login]



38

ACL incoming traffic

- 17. If TCP destination port = 22, DENY
  [SSH for secure login, but some versions are insecure]
- 18. If UDP destination port=69, DENY
  [Trivial File Transfer Protocol; no login necessary]
- 19. If ICMP Type = 0, PASS
  [allow incoming echo reply messages]
- **20. DENY ALL**



39

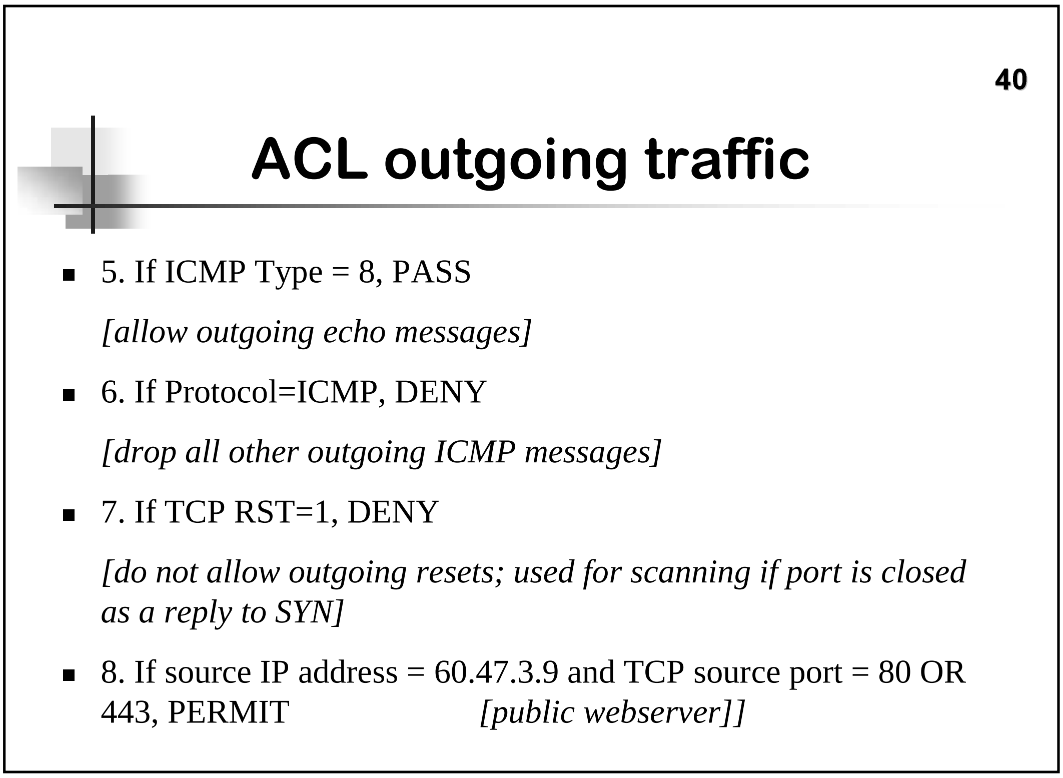
ACL outgoing traffic

- 1. If source IP address = 10.*.*.*, DENY

  [private IP address range]
- 2. If source IP address = 172.16.*.* to 172.31.*.*, DENY

  [private IP address range]
- 3. If source IP address = 192.168.*.*, DENY

  [private IP address range]
- 4. If source IP address NOT = 60.47.*.*, DENY
- *[not in internal address range]*



40

ACL outgoing traffic

- 5. If ICMP Type = 8, PASS

  [allow outgoing echo messages]
- 6. If Protocol=ICMP, DENY

  [drop all other outgoing ICMP messages]
- 7. If TCP RST=1, DENY

  [do not allow outgoing resets; used for scanning if port is closed as a reply to SYN]
- 8. If source IP address = 60.47.3.9 and TCP source port = 80 OR 443, PERMIT *[public webserver]]*

41

ACL outgoing traffic

- 9. If TCP source port=0 through 49151, DENY

  [well-known and registered ports]
- 10. If UDP source port=0 through 49151, DENY

  [well-known and registered ports]
- 11. If TCP source port =49152 through 65,536, PASS

  [allow outgoing client connections]
- 12. If UDP source port = 49152 through 65,536, PERMIT

  [allow outgoing client connections]

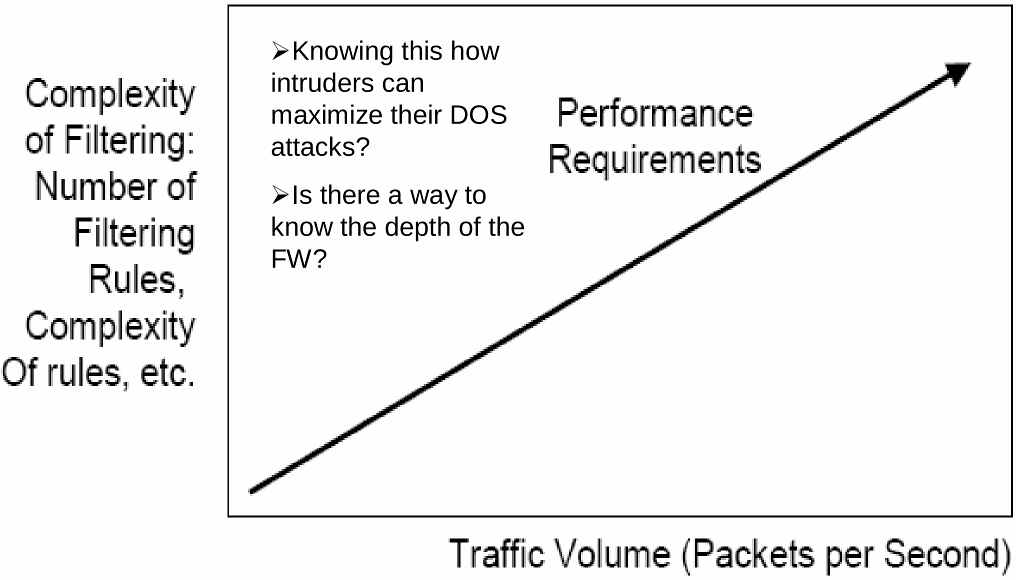
13. **DENY ALL**

42

ACL

- Beware of misconfiguration
- Rules of subset, superset, overlapping, shadowing (one rule never gets triggered).

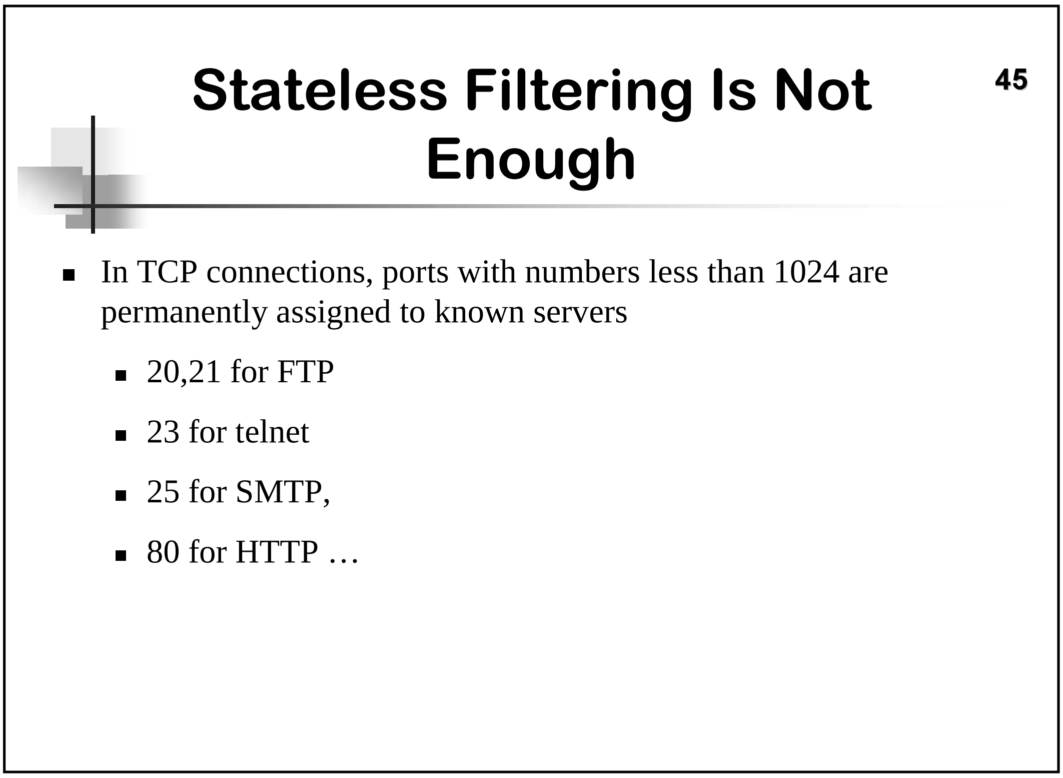
Firewall Performance



- ➢Knowing this how intruders can maximize their DOS attacks?
- ➢Is there a way to know the depth of the FW?

Weaknesses of Packet Filters

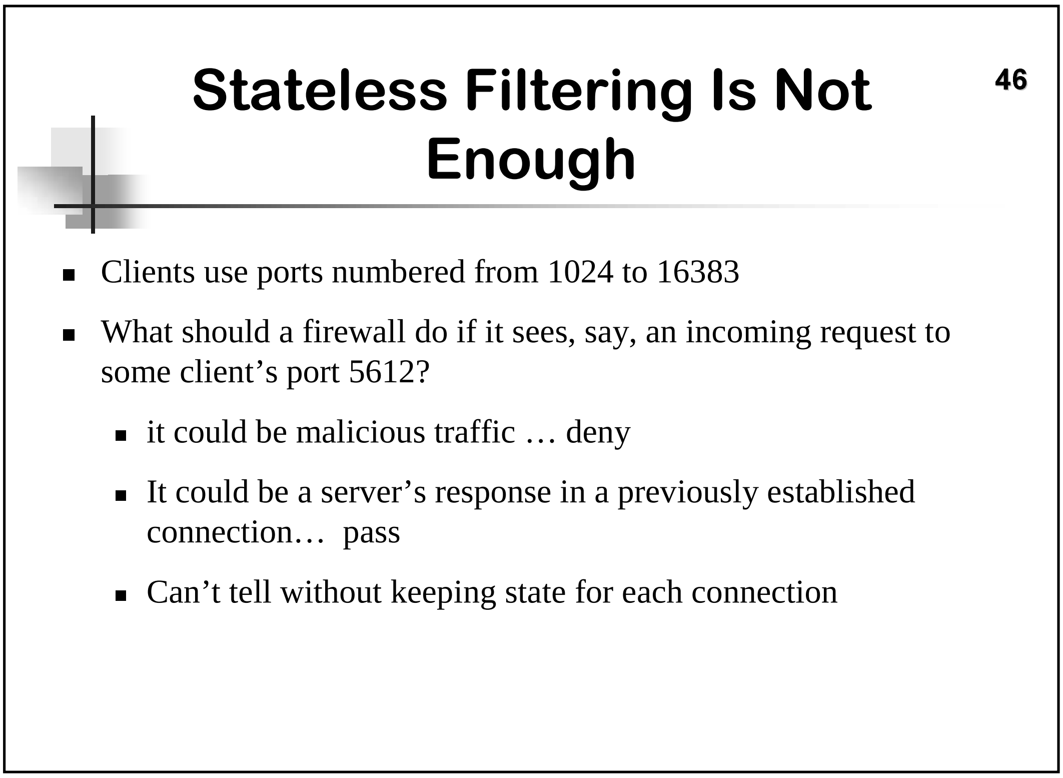
- Do not prevent application-specific attacks
- No user authentication mechanisms:
  - Address-based authentication is very weak as it can be spoofed!
  - Firewalls don’t have any upper-level functionality
- Vulnerable to TCP/IP attacks such as spoofing
- Does not know how to deal with returning traffic, usually has ephemeral ports!



Stateless Filtering Is Not Enough

45

- In TCP connections, ports with numbers less than 1024 are permanently assigned to known servers
  - 20,21 for FTP
  - 23 for telnet
  - 25 for SMTP,
  - 80 for HTTP …



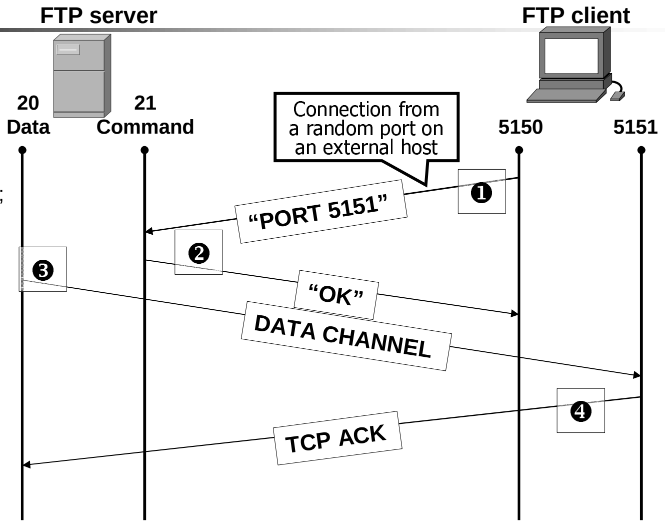
Stateless Filtering Is Not Enough

46

- Clients use ports numbered from 1024 to 16383
- What should a firewall do if it sees, say, an incoming request to some client's port 5612?
  - it could be malicious traffic … deny
  - It could be a server's response in a previously established connection… pass
  - Can't tell without keeping state for each connection

Example: FTP

47



❶ Client opens command channel to server; tells server second port number

❷ Server acknowledges

❸ Server opens data channel to client's second port

❹ Client acknowledges

48

Session Filtering

- Decision is still made separately for each packet, but in the context of a connection
  - If new connection, then check against security policy
  - If existing connection, then look it up in the table and update the table, if necessary
    - Only allow incoming traffic to a high-numbered port if there is an established connection to that port
  - FASTER than packet Filtering – does not check the Rulebase if Established/Related!

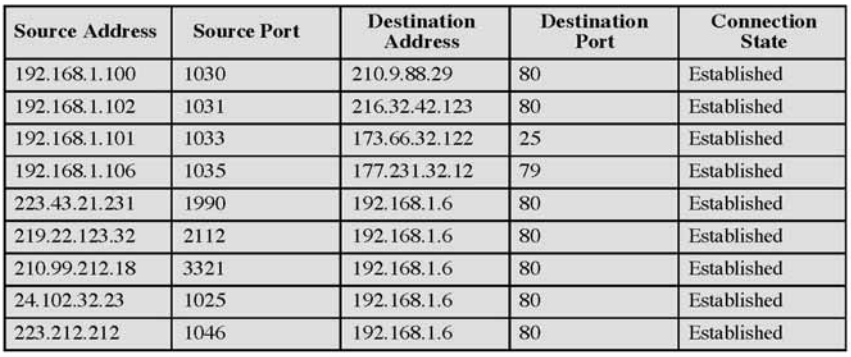
49

Session Filtering

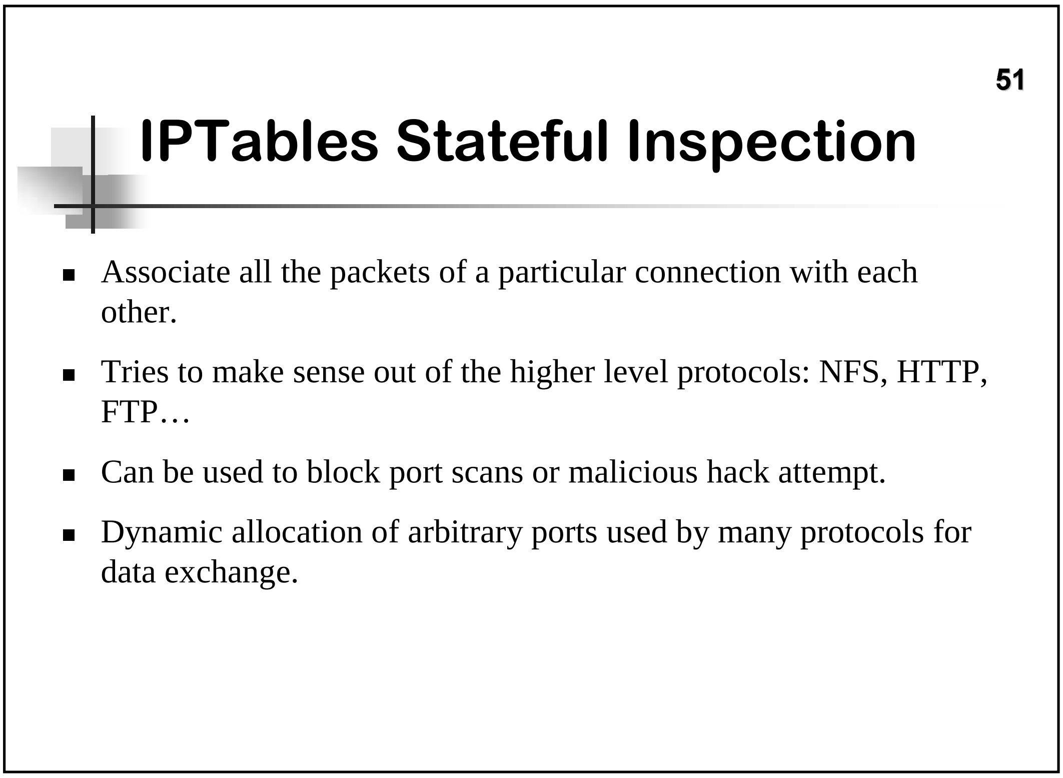
- Hard to filter stateless protocols (UDP) and ICMP
  - Stateful is not faster for UDP traffic!!
  - That is why better to have two FWs back-to-back.
    - Stateless FW to filter noisy traffic (UDP traffic)
    - Followed by a Stateful FW to deal with TCP filtering
- Typical filter: deny everything that’s not allowed
  - Must be careful filtering out service traffic such as ICMP
- Filters can be bypassed with IP tunneling

50

Example: Connection State Table



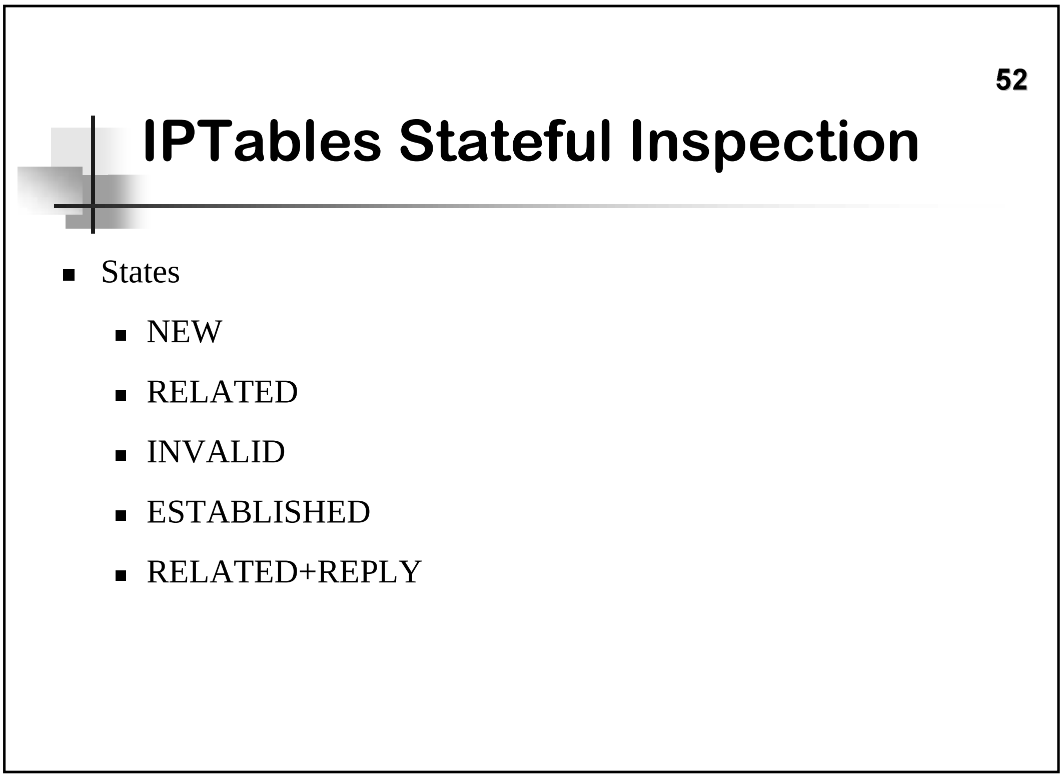
Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.9.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	80	Established
219.22.123.32	2112	192.168.1.6	80	Established
210.99.212.18	3321	192.168.1.6	80	Established
24.102.32.23	1025	192.168.1.6	80	Established
223.212.212	1046	192.168.1.6	80	Established



51

IPTables Stateful Inspection

- Associate all the packets of a particular connection with each other.
- Tries to make sense out of the higher level protocols: NFS, HTTP, FTP…
- Can be used to block port scans or malicious hack attempt.
- Dynamic allocation of arbitrary ports used by many protocols for data exchange.



52

IPTables Stateful Inspection

- States
  - NEW
  - RELATED
  - INVALID
  - ESTABLISHED
  - RELATED+REPLY

53

Available Firewalls

- Buy a solution
  - Hardware -- PIX, Sonicwall, WatchGuard…
  - Software -- CheckPoint, ISA, Boarder Manager
- Build a solution
  - Linux -- IPTables
  - BSD -- IPFW, IPFilter, pf

54

Protecting Addresses and Routes

- Hide IP addresses of hosts on internal network
  - Only services that are intended to be accessed from outside need to reveal their IP addresses
  - Keep other addresses secret to make spoofing harder
- Use NAT (network address translation) to map addresses in packet headers to internal addresses
  - 1-to-1 or N-to-1 mapping

Protecting Addresses and Routes

55

- Filter route announcements
  - No need to advertise routes to internal hosts
  - Prevent attacker from advertising that the shortest route to an internal host lies through him

General Problems with Firewalls

56

- Interfere with networked applications
- Don’t solve real problems
  - Buggy software (think buffer overflow exploits)
  - Bad protocol design (think WEP in 802.11b)
  - Don’t prevent insider attacks
- Generally don’t prevent denial of service
- Increasing complexity and potential for misconfiguration