Practical Network Defense

Master's degree in Cybersecurity 2019-20

Network traffic regulation with firewalls

Angelo Spognardi
spognardi@di.uniroma1.it
Dipartimento di Informatica
Sapienza Università di Roma

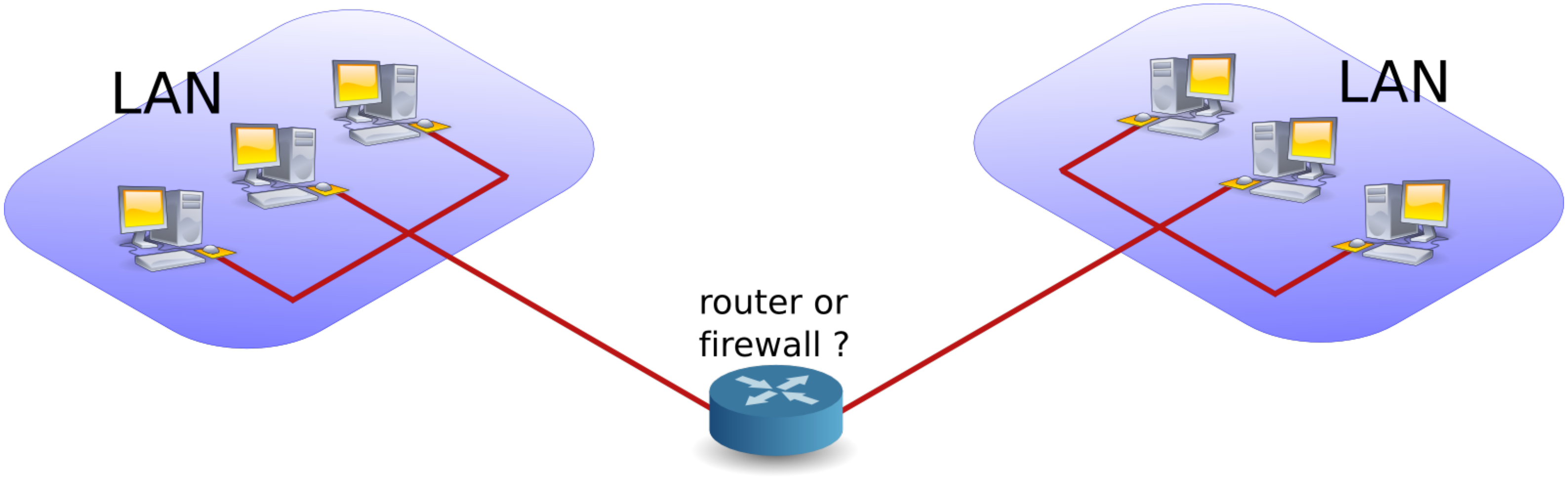
Today's agenda

- Traffic regulation
  - Packet filtering
- Filter rules
- Stateful firewall
- Other types of firewall
  - Application-level filtering
  - Circuit-level gateway

Traffic regulation

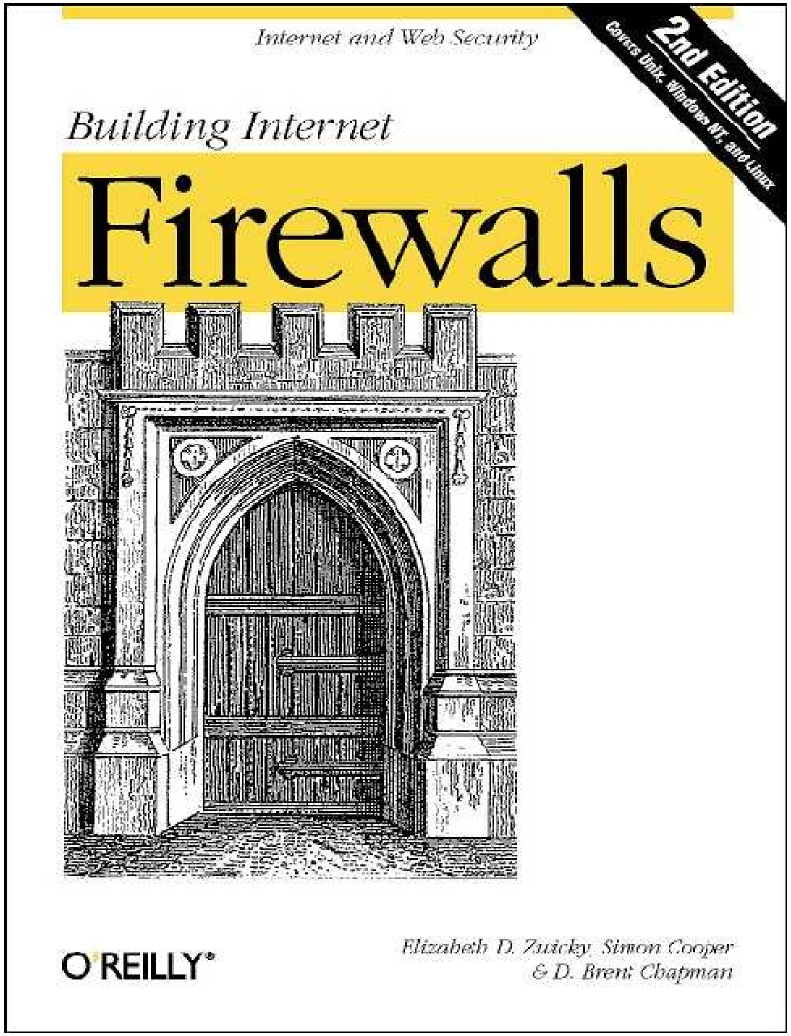
Regulate traffic: routers and firewalls

- A router is a device that connects two networks
- A firewall is a device that besides acting as a router, also contains (and implements) rules to determine whether packets are allowed to travel from one network to another
  - Router also can perform some form of screening (packet filter)



Why firewalls?

- Restricts access from the outside
  - Internet = millions of people together → bad things happen
- Prevents attackers from getting too close
- Restricts people from leaving


Internet and Web Security
2nd Edition
Covers Unix, Windows NT, and Linux
Building Internet
Firewalls
O'REILLY®
Elizabeth D. Zwicky, Simon Cooper
& D. Brent Chapman

Secure traffic regulation

- To attain a certain level of network security, you can:
  - Regulate which traffic is allowed (sources, destinations, services, …)
  - Protect the traffic by encryption
  - Monitor the traffic for “bad behaviour”
  - Monitor the hosts for “bad behaviour”
- The choice will depend on the security policy to be fulfilled (particularly the CIA targets).

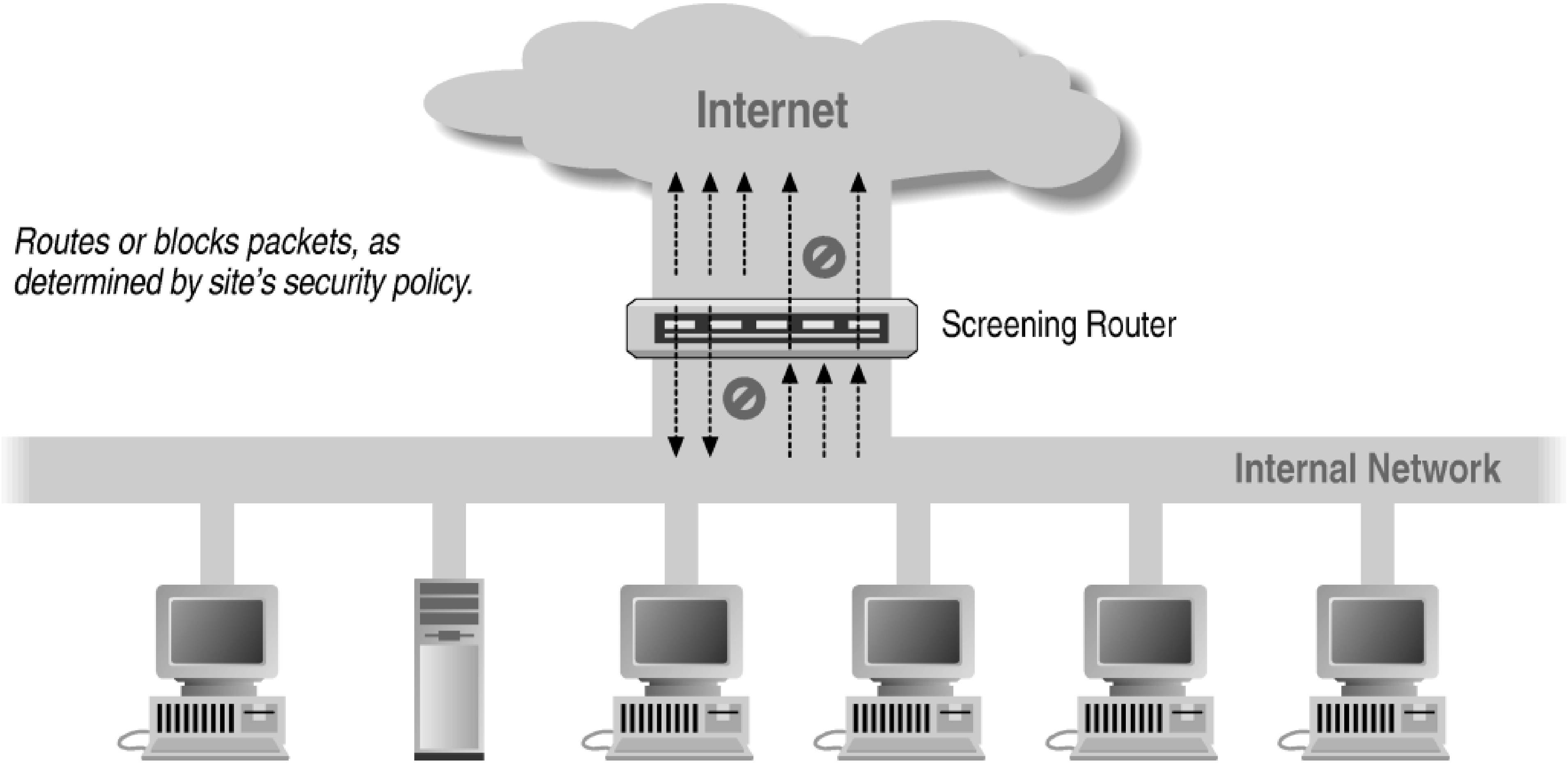
Firewall Design & Architecture Issues

- Least privilege
- Defense in depth
- Choke point
- Weakest links
- Fail-safe stance
- Universal participation
- Diversity of defense
- Simplicity

Host based packet filters

- Kind of firewall that disciplines the traffic in/out a single host
- It specifies the packets that can be received and sent
  - Ex: iptables, windows firewall and all the so called "personal firewalls"
- Vendor products generally work per-app: each installed application has a known policy that has to obey

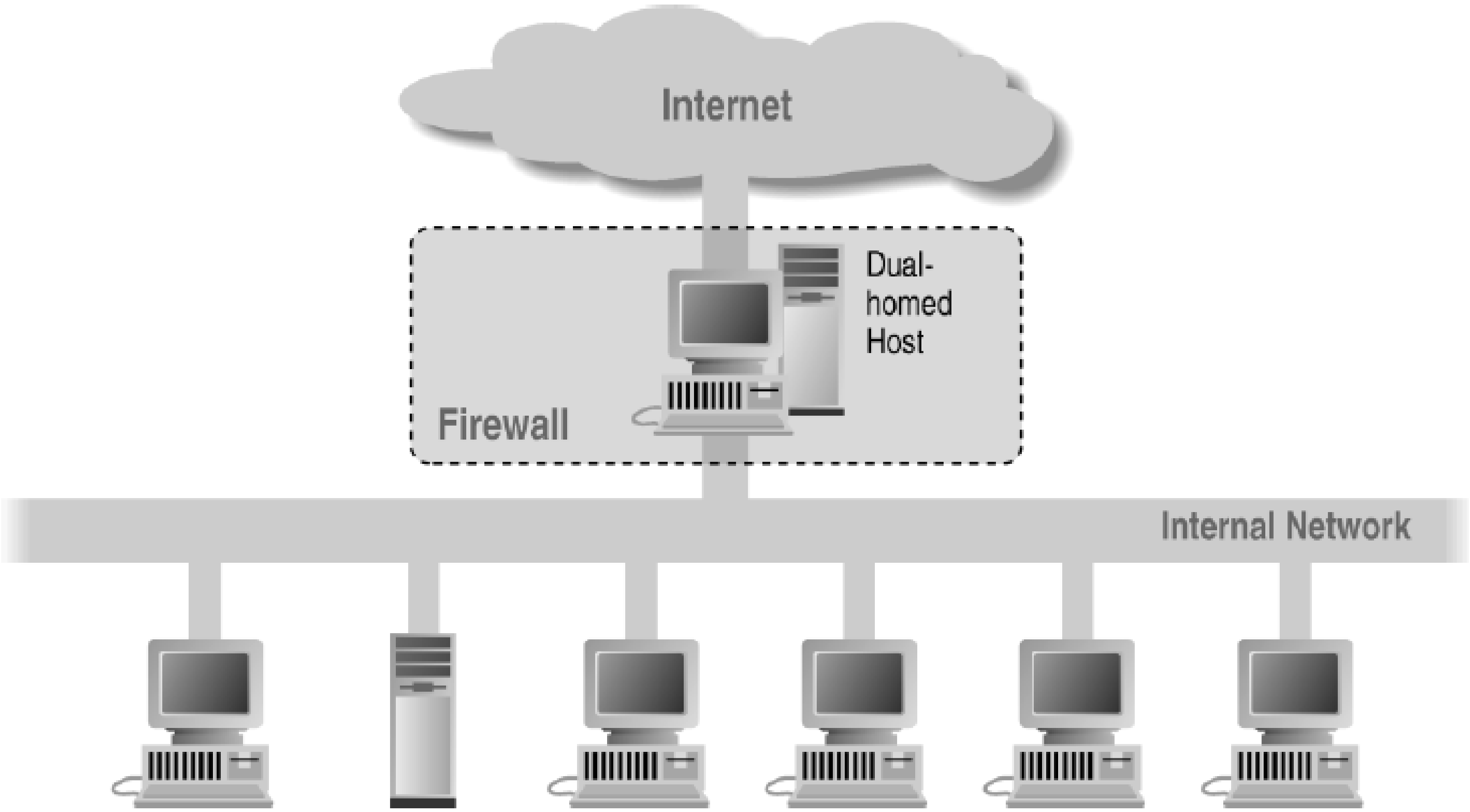
Screening router (ACL-based)


Internet
Routes or blocks packets, as determined by site's security policy.
Screening Router
Internal Network

Network Access Control Lists

- List the rights for accessing/using networks
  - Extensively used in switches, routers and firewalls
- Usually distinguish between incoming and outgoing traffic, per interface/port
  - Ex: lists of IP addresses that can send packets to an interface/port
- Stateless: every packet is treated independently, without any knowledge of what has come before

Dual-homed host



Bastion host

- Hardened computer used to deal with all traffic coming to a protected network from outside
  - Hardening is the task of reducing or removing vulnerabilities in a computer system:
    - Shutting down unused or dangerous services
    - Strengthening access controls on vital files
    - Removing unnecessary accounts and permissions
    - Using "stricter" configurations for vulnerable components, such as DNS, sendmail, FTP, Apache, Tomcat, etc.
- Specially suitable for use as Application Proxy Gateways

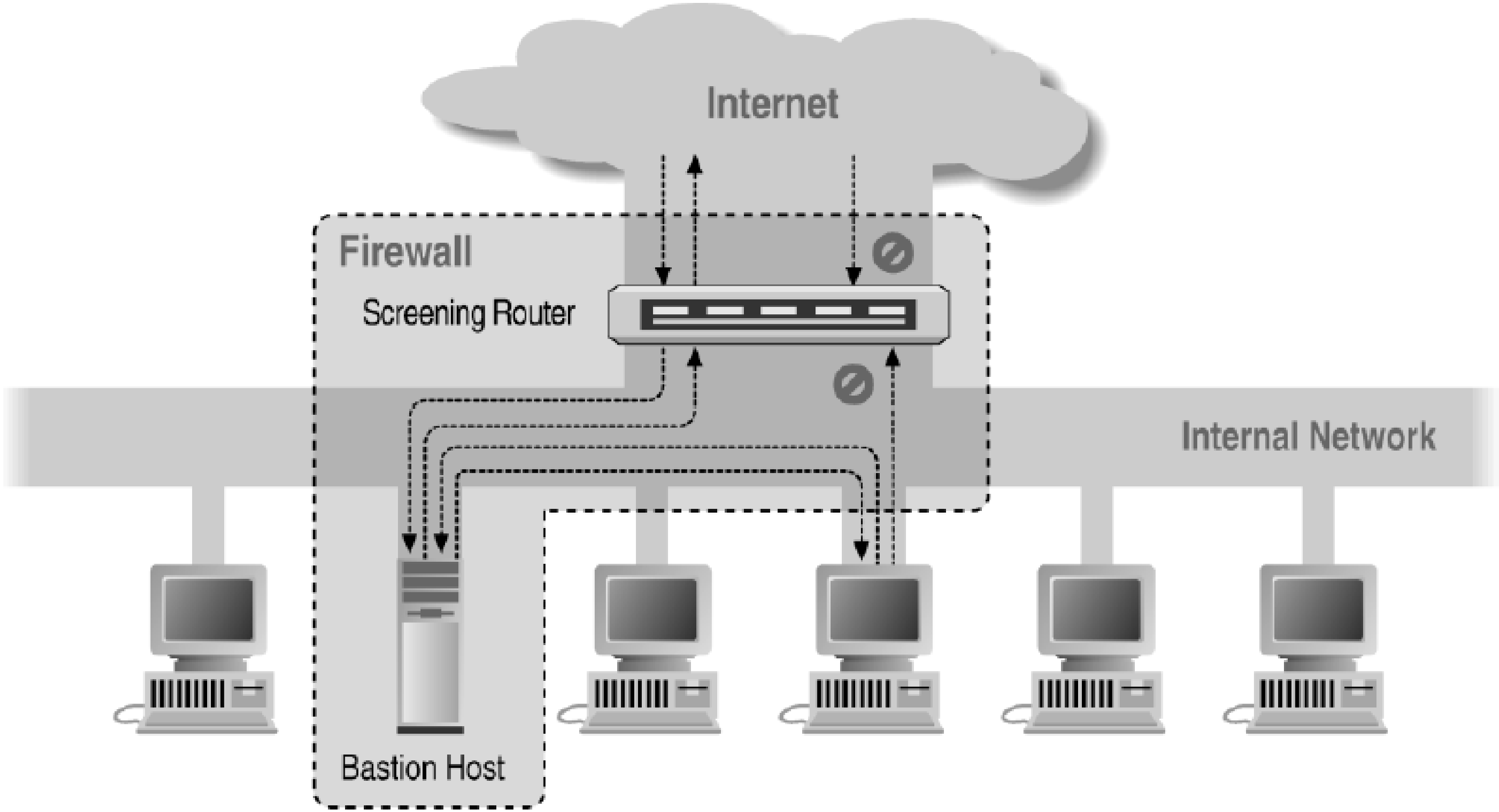
What is a DMZ

- DMZ (demilitarized zone)
  - Computer host or small network inserted as a “neutral zone” between a company’s private network and the outside public network
  - Network construct that provides secure segregation of networks that host services for users, visitors, or partners
- DMZ use has become a necessary method of providing a multilayered, **defense-in-depth** approach to security
- Reduce and regulate the access to internal (private) components of the IT system

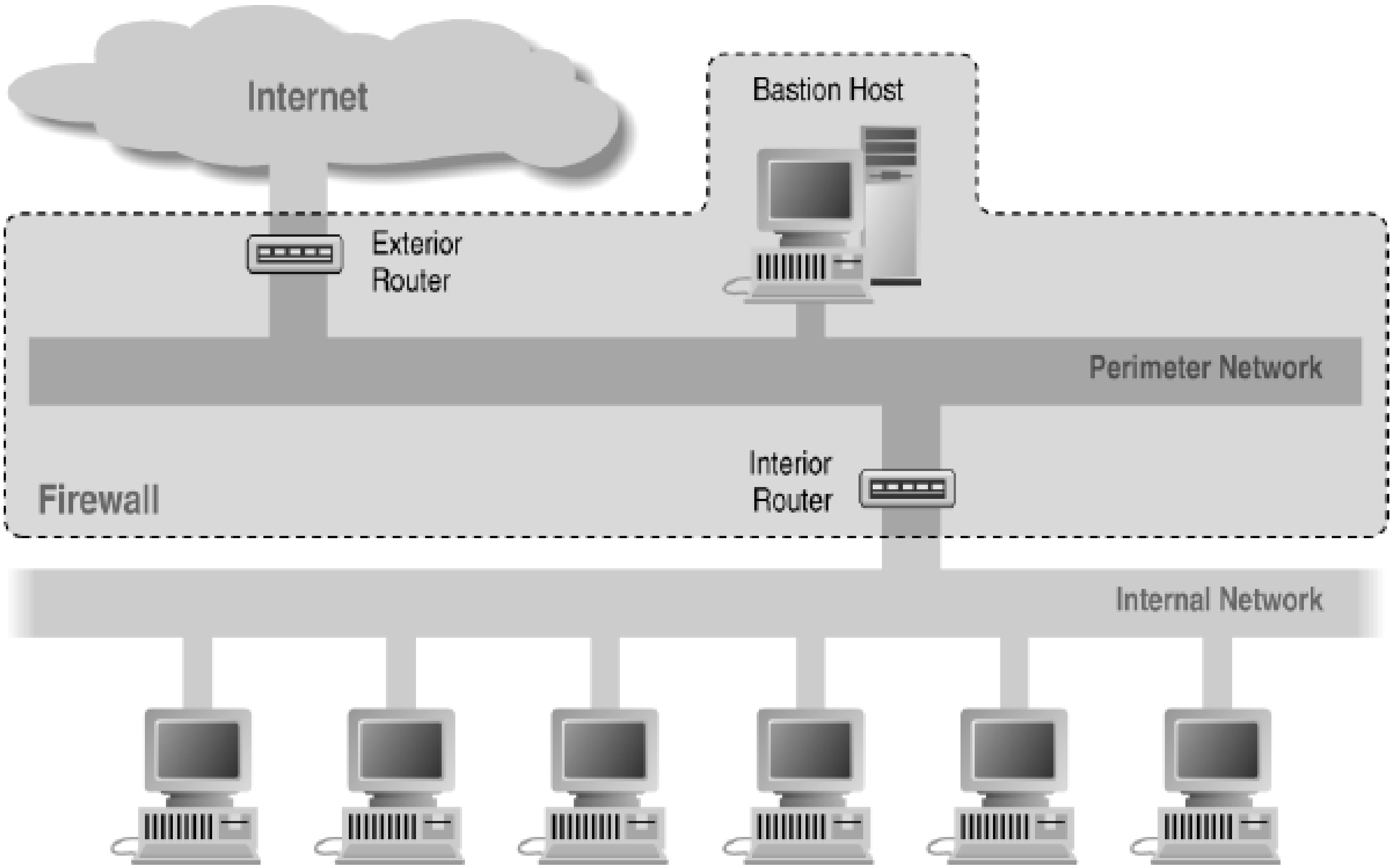
Defense in depth

- A security approach in which IT systems are protected using **multiple overlapping** systems
  - Add redundancy to the defensive measures
  - Aim to remove the single point of failure
  - Find the right balance between complexity and multiplicity of defense measures
- In order to compromise the system, an attacker has to find multiple vulnerabilities, in different components

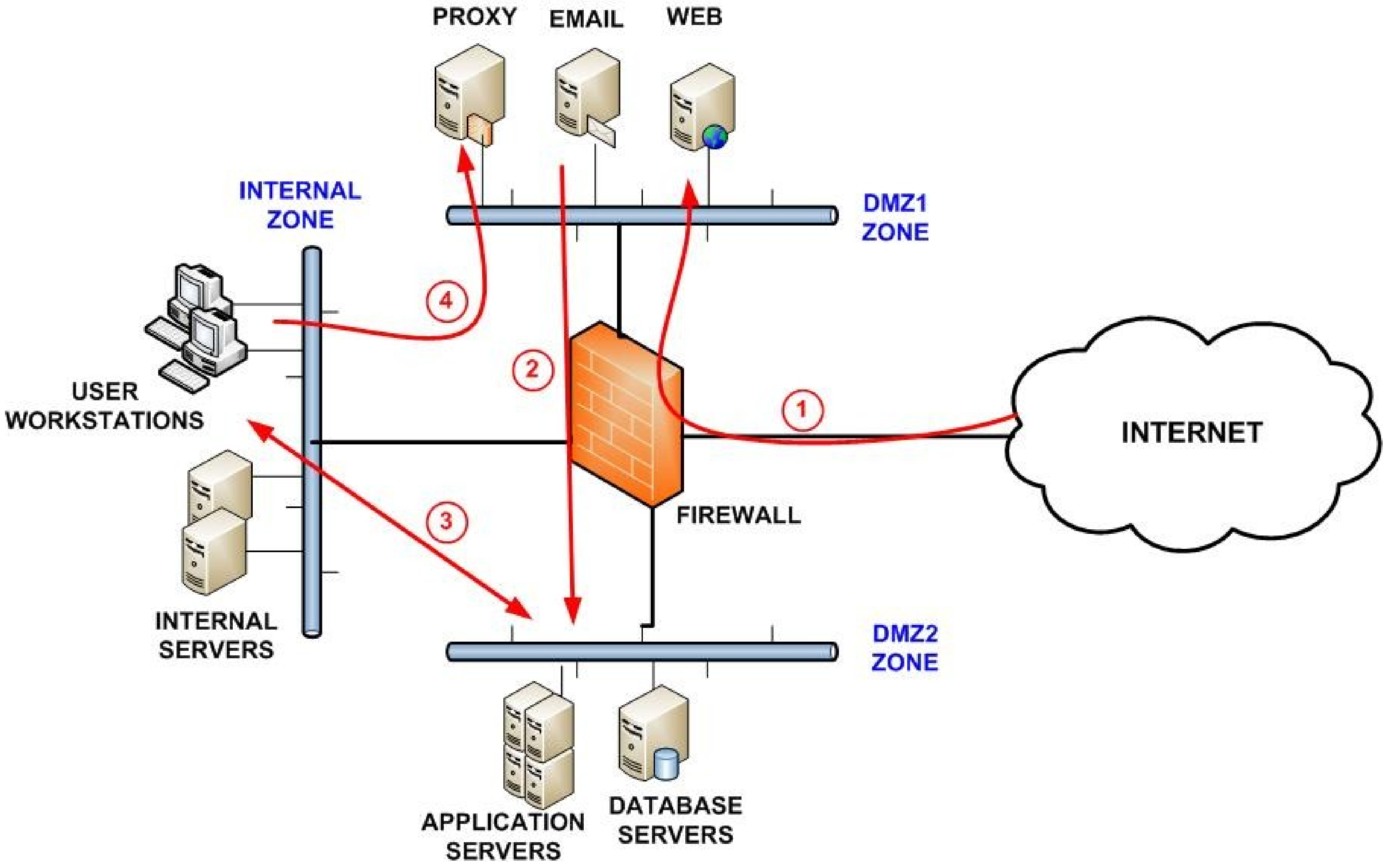
DMZ as a screened Host



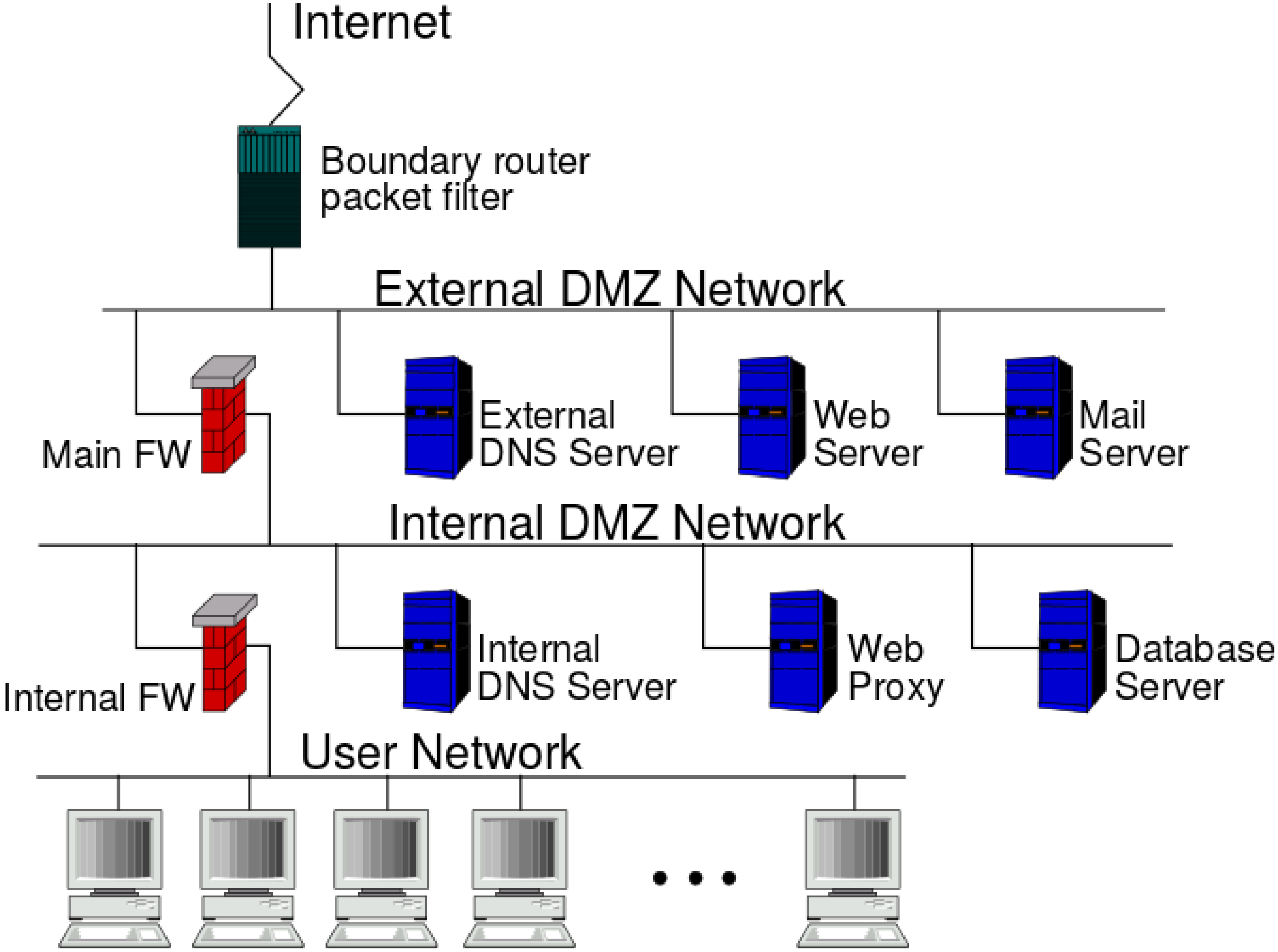
Screened Subnet Using Two Routers/Firewalls



DMZ to segment the network

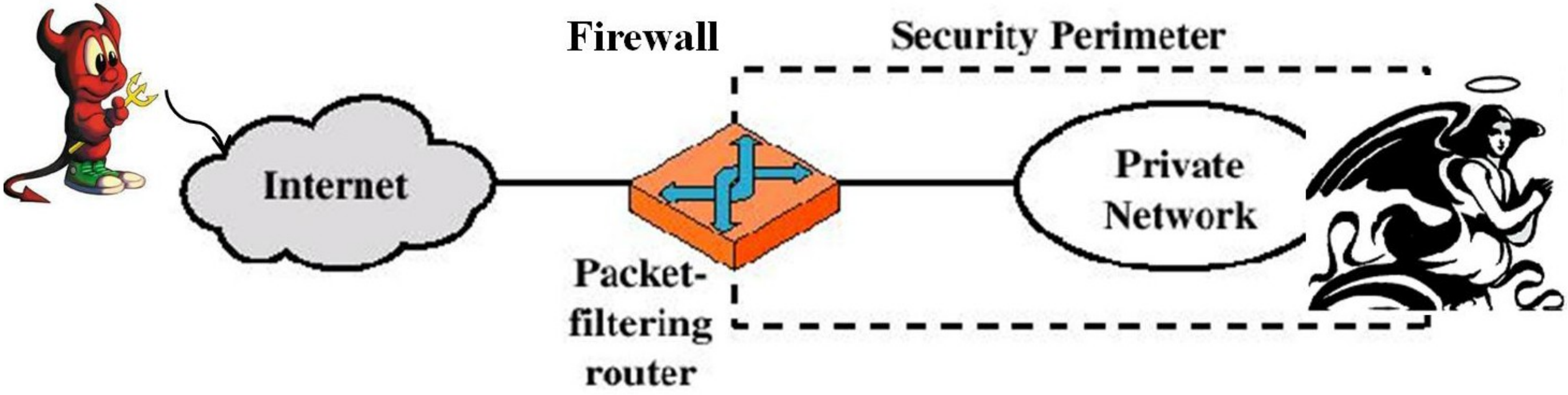


Security in depth: split DMZ



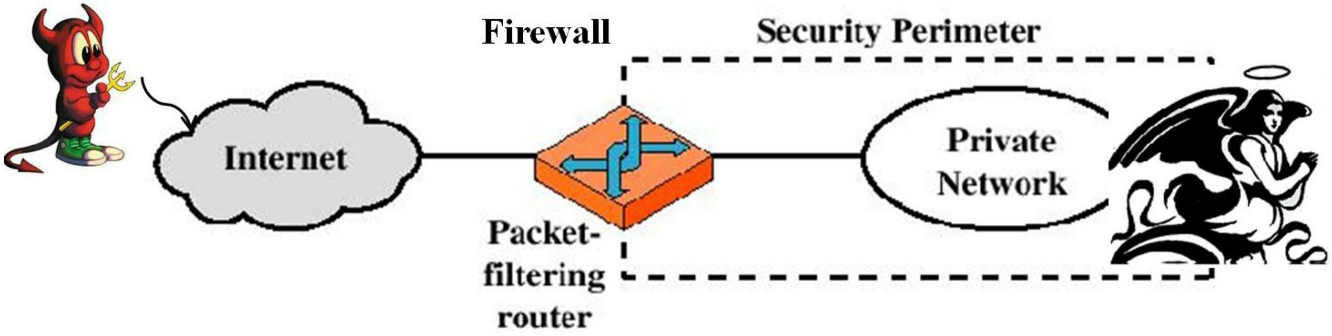
A simple plan for network security

- Use a firewall to filter ingoing and outgoing traffic between “your” network (or individual PC) and the Internet



Assumptions

1. You have security policy stating what is allowed and not allowed.
2. You can identify the “good” and the “bad” traffic by its IP-address, TCP port numbers, etc, …
3. The firewall itself is immune to penetration.
   - A question of assurance – needs for a trusted system, secure OS etc.

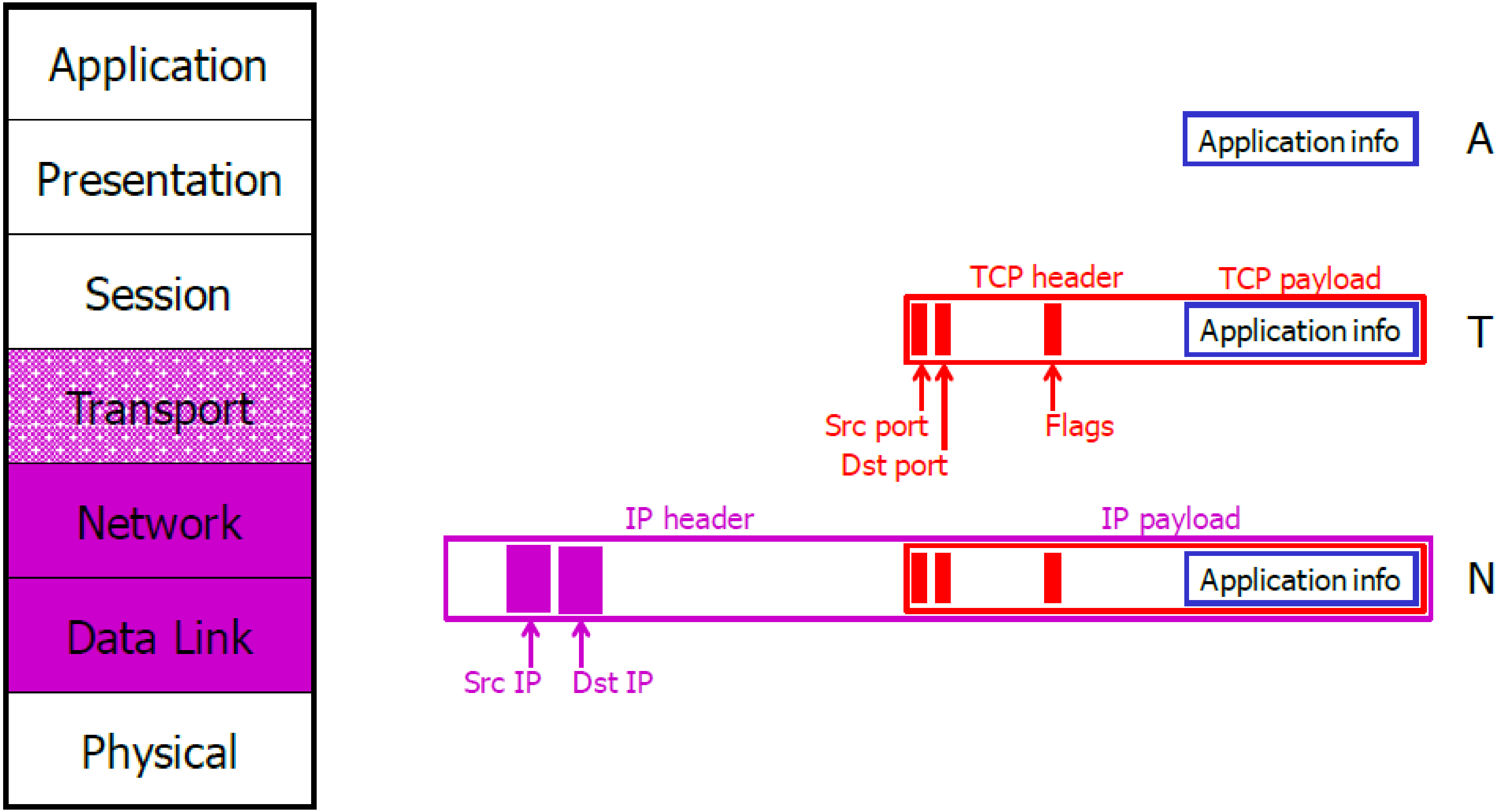


Packet filters (stateless firewall)

- Drop packets based on their source or destination addresses or port numbers or flags
- No context, only contents
- Can operate on
  - incoming interface
  - outgoing interface
  - both

- Check packets with fake IP addresses:
  - from outside ("ingress filtering")
  - from inside ("egress filtering")

Packet filters operating layers



Three-step process

1. Know your policy
2. Translate the policy in a formal language
    - E.g.: logical expression on packet fields
3. Rewrite the policy in terms of the firewall syntax

General mechanism:

- Rules are checked from top to bottom
- The first matching rule is applied
- One implicit rule is assumed if no rule matches
    - Block/Allow everything

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

Example

- Policy:
  - allow inbound email (SMTP, port 25) only to our-gateway machine: **`Mailgw`**
  - refuse all traffic from a known spamming site: **`demon`**
- Possible rules:

action	ourhost	port	theirhost	port	comment
block	*	*	demon	*	*don't trust spammers*
allow	Mailgw	25	*	*	*connection to our SMTP*

Example, continued

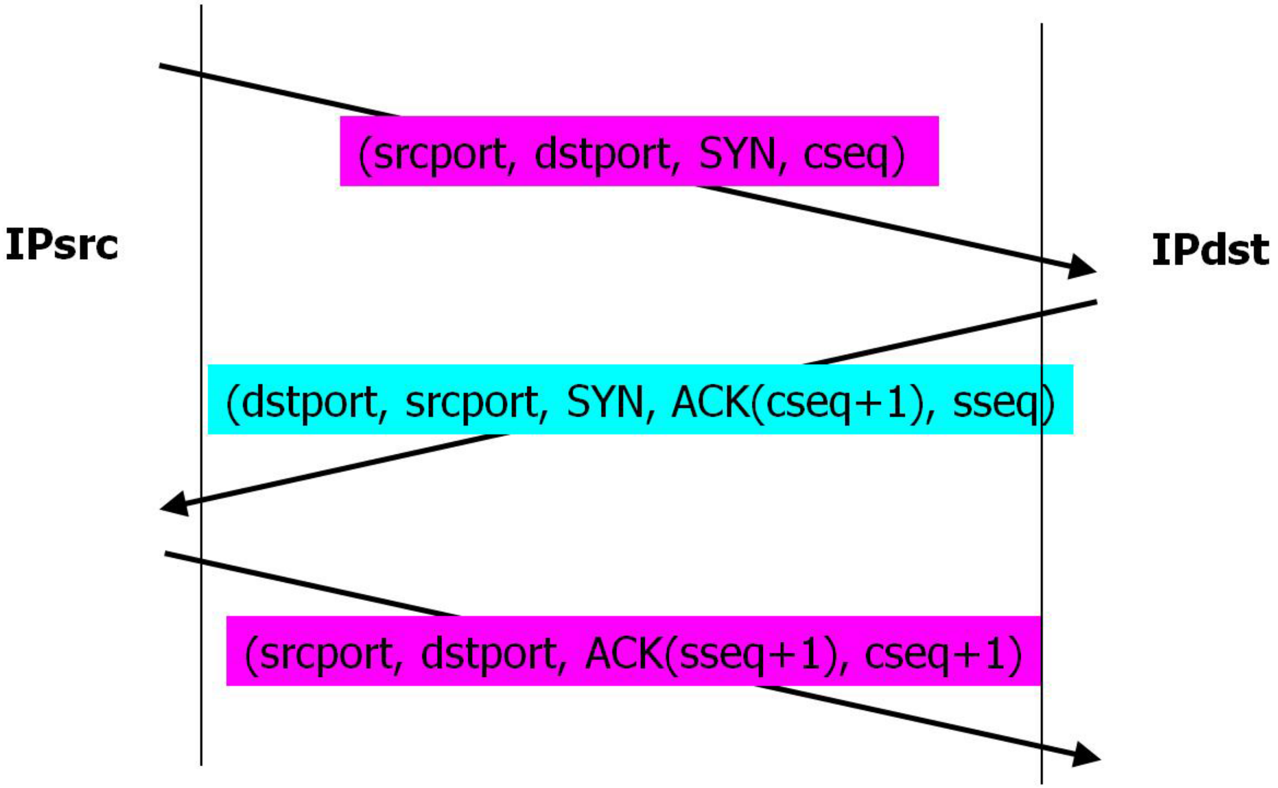
- Add the policy:
  - any inside host can send mail to the outside

action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	*connection to their SMTP*

- Very bad: we can not control the type of traffic originated from port 25 and coming from the outside
- Then: rules have to specify the direction of the traffic

How to check the direction of TCP?

- Consider the TCP flags


IPsrc
IPdst
(srcport, dstport, SYN, cseq)
(dstport, srcport, SYN, ACK(cseq+1), sseq)
(srcport, dstport, ACK(sseq+1), cseq+1)

Example with traffic direction

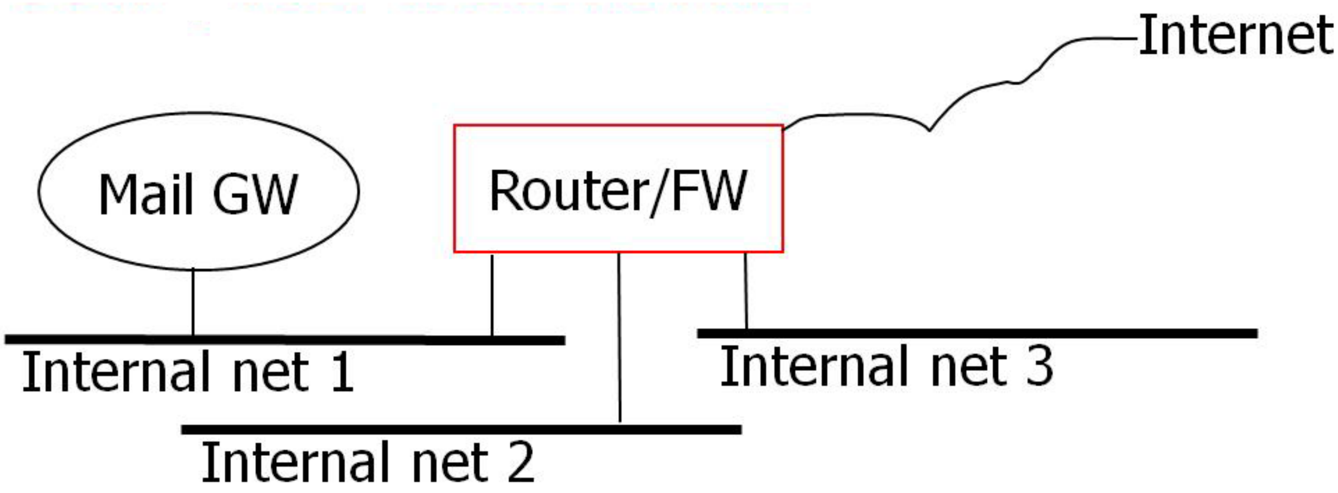
- We distinguish the replies to our SMTP connection considering the ACK flag

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	25		*connection to their SMTP*
allow	*	25	*	*	ACK	*their replies*
block	*	*	*	*		*default*

- Very easy case…

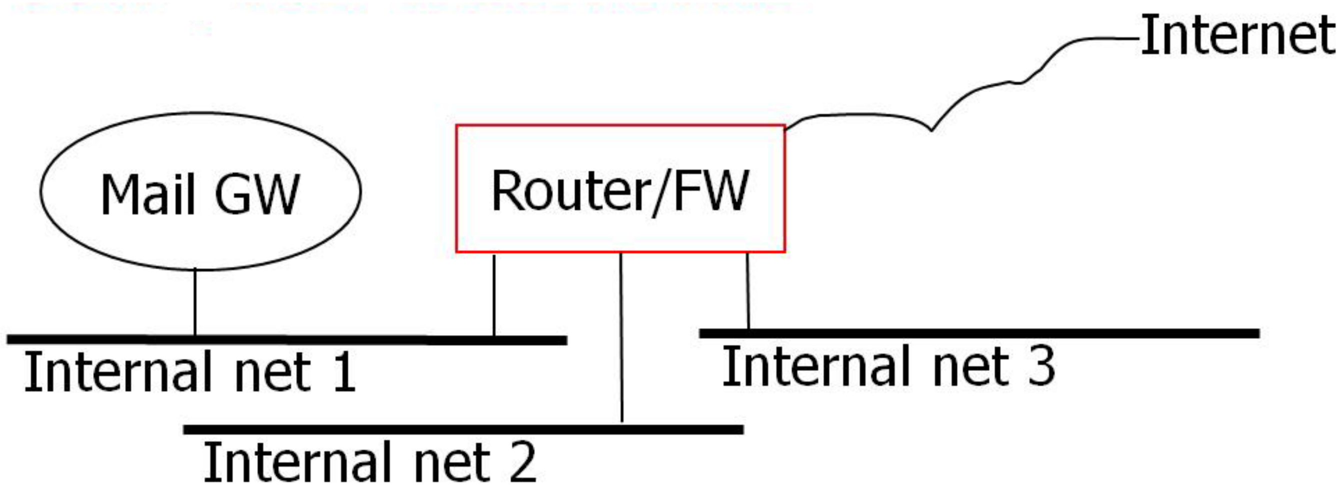
Filter rules for network firewalls

More complex network topology



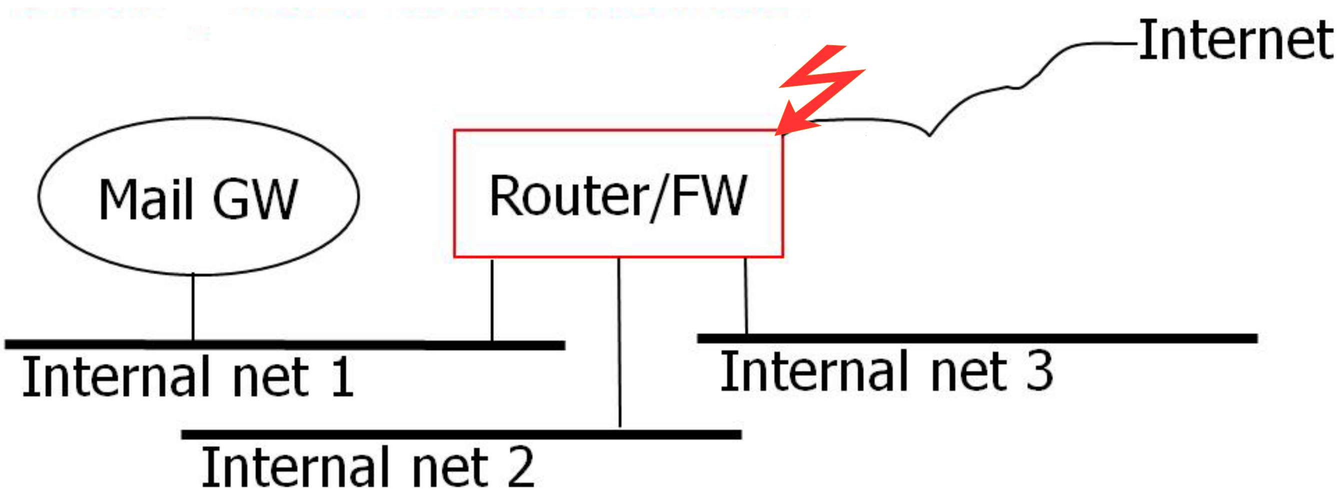
- Policy:
  - Internal Net 1 is a DMZ and only hosts Mail GW
  - Very limited connections between Mail GW and Internet (only partner servers)
  - Limited connections allowed between Mail GW and net 2 and net 3
  - Anything can pass between net 2 and net 3
  - Outgoing requests only between net 2 or net 3 and the link to the Internet

Requirements



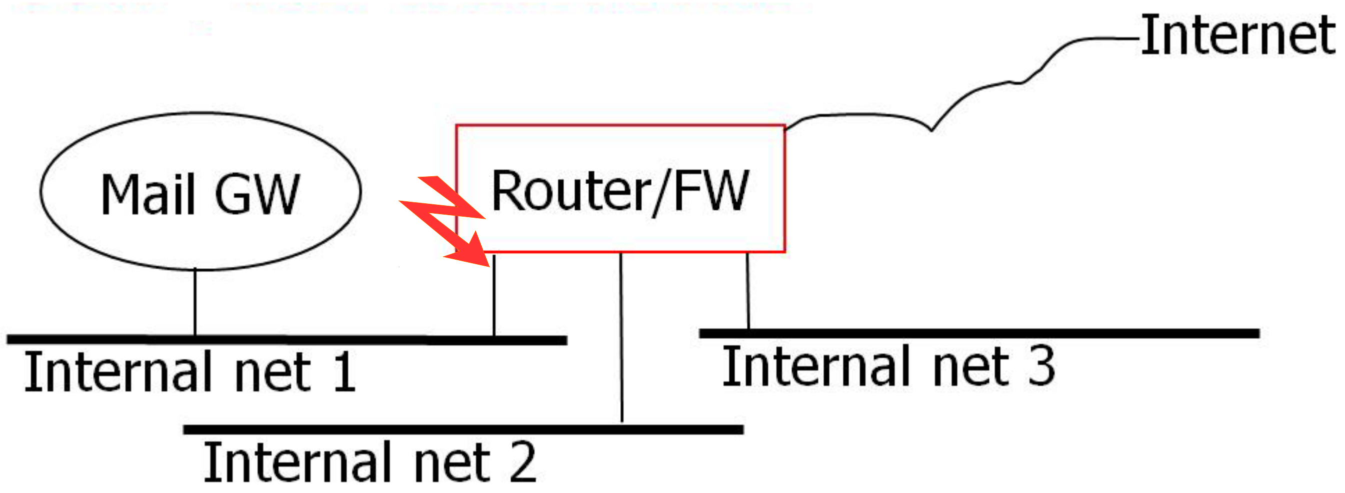
- We cannot only consider where packets have to go (destination→ **egress filtering**)
  - Open access to net 2 only allowed for traffic with source address in net 3
  - No way to avoid fake source addresses (*address spoofing*) from outside
- We need to define rules based on **from where packets are arriving,** (source → **ingress filtering**)

Interface towards Internet



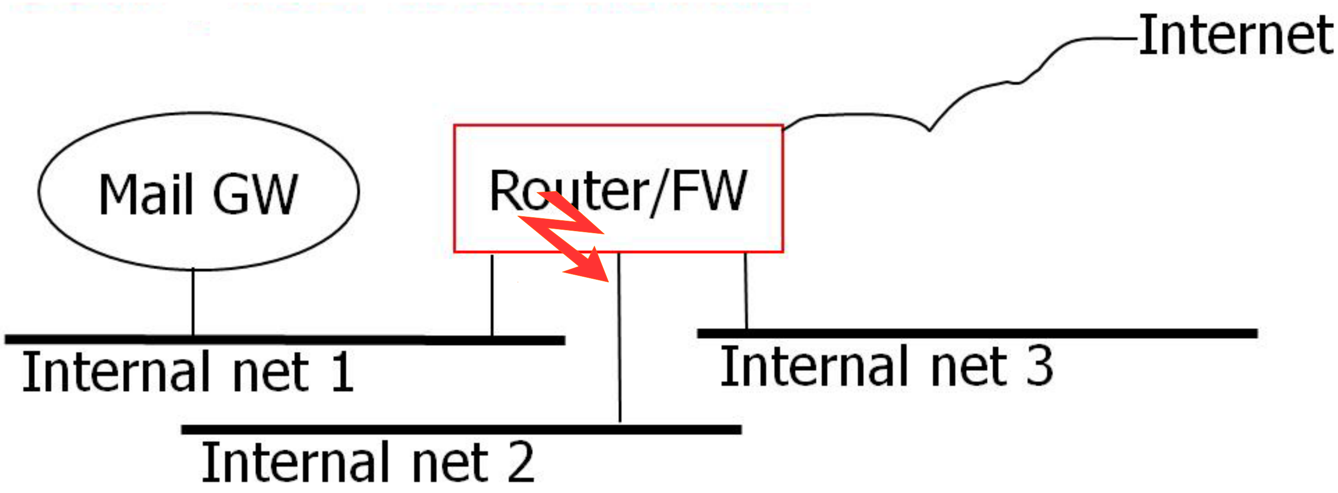
Action	IPsrc	srcport	IPdst	dstport	flags
block	"Net 1"	*	*	*	
block	"Net 2"	*	*	*	
block	"Net 3"	*	*	*	
allow	*	*	GW	25	
allow	*	*	"Net 2"	*	ACK
allow	*	*	"Net 3"	*	ACK

Interface on net 1



Action	IPsrc	srcport	IPdst	dstport	flags
allow	GW	*	"partners"	25	
allow	GW	*	"Net 2"	*	ACK
allow	GW	*	"Net 3"	*	ACK
block	GW	*	"Net 2"	*	
block	GW	*	"Net 3"	*	
allow	GW	*	*	*	

Interface on net 2 (net 3 is similar)



action	IPsrc	srcport	IPdst	dstport	flags
allow	"Net 2"	*	*	*	
block	*	*	*	*	

Problems with Packet Filters

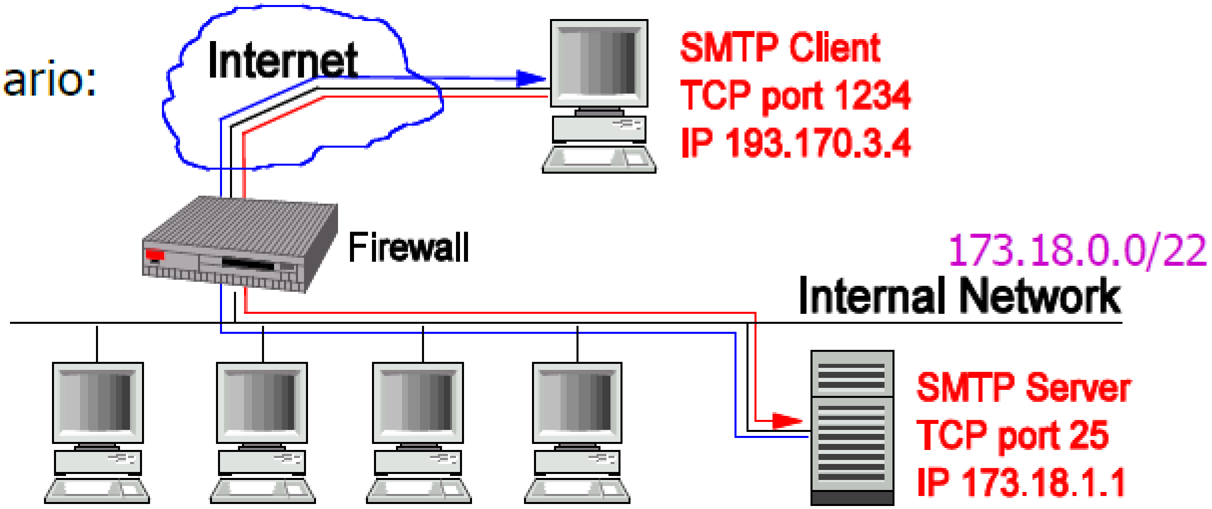
- Only a small number of parameters
  - it is (unfortunately) easy to specify filtering rules which are too specific or too general
- Payload of TCP packet is not inspected
  - No protection against attacks based on upper-layer vulnerabilities
- Limited logging ability (restricted to the few parameters used by the filter)
- No authentication facilities
- Susceptible to attacks based on vulnerabilities in various implementations of TCP and/or IP

Filter rules, 1

- Example: Filtering in- and outgoing SMTP traffic. Try rules:

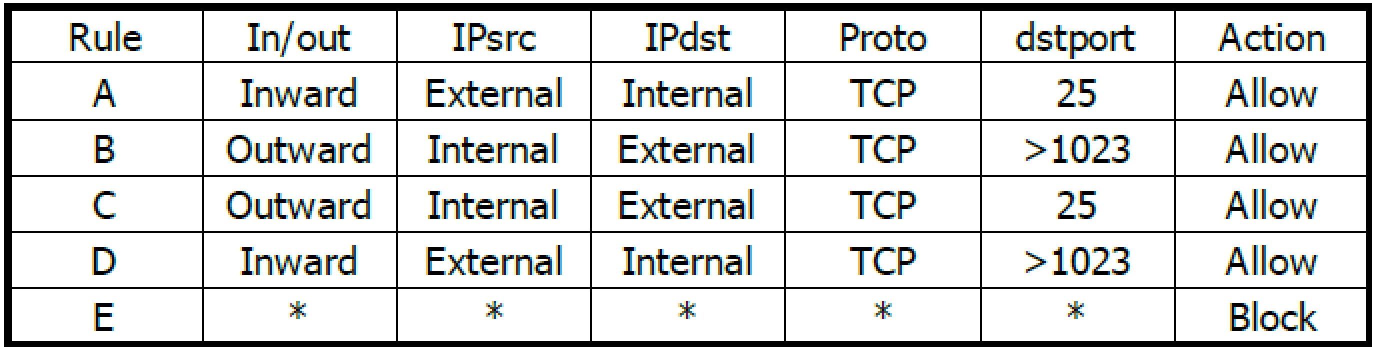
Rule	In/out	IPsrc	IPdst	Proto	dstport	Action
A	Inward	External	Internal	TCP	25	Allow
B	Outward	Internal	External	TCP	>1023	Allow
C	Outward	Internal	External	TCP	25	Allow
D	Inward	External	Internal	TCP	>1023	Allow
E	*	*	*	*	*	Block

- Scenario:



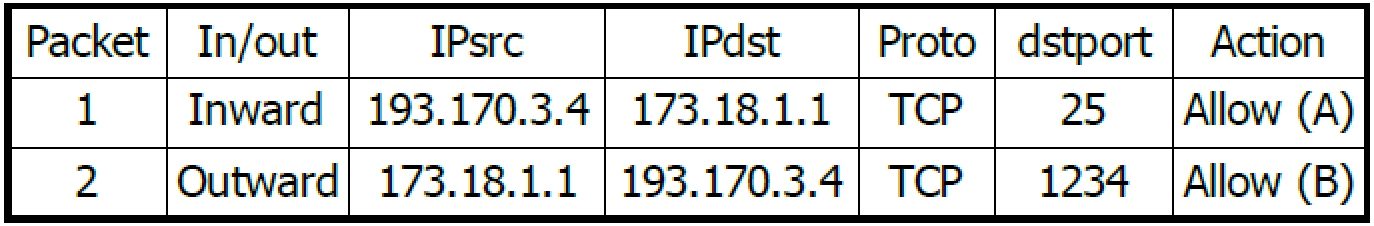
Filter rules, 2

- Example: Filtering in- and outgoing SMTP traffic. Try rules:



Rule	In/out	IPsrc	IPdst	Proto	dstport	Action
A	Inward	External	Internal	TCP	25	Allow
B	Outward	Internal	External	TCP	>1023	Allow
C	Outward	Internal	External	TCP	25	Allow
D	Inward	External	Internal	TCP	>1023	Allow
E	*	*	*	*	*	Block

- Scenario:



Packet	In/out	IPsrc	IPdst	Proto	dstport	Action
1	Inward	193.170.3.4	173.18.1.1	TCP	25	Allow (A)
2	Outward	173.18.1.1	193.170.3.4	TCP	1234	Allow (B)

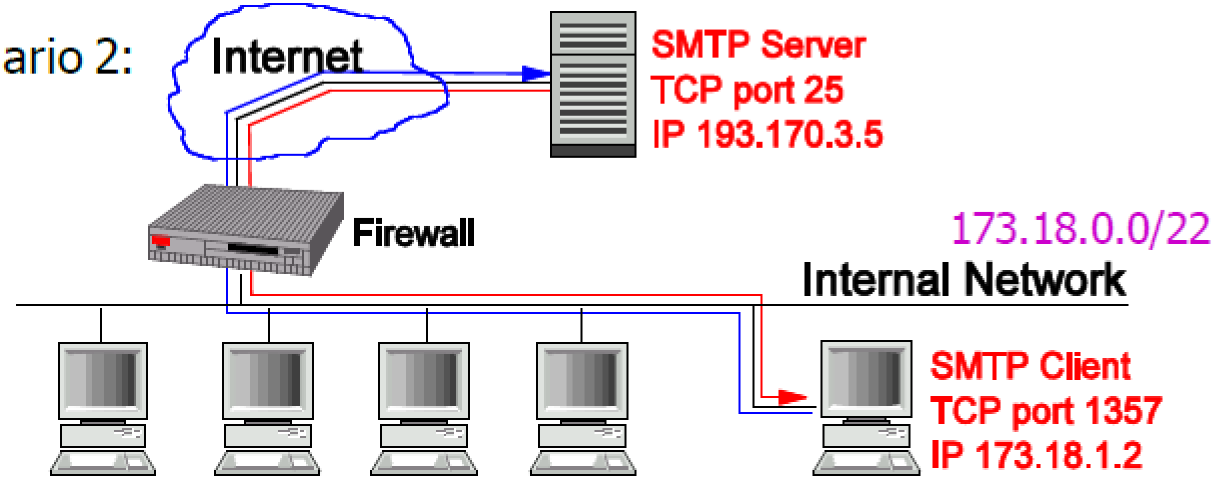
- Conclusion: This looks OK!

Filter rules, 3

- Example: Filtering in- and outgoing SMTP traffic. Try rules:

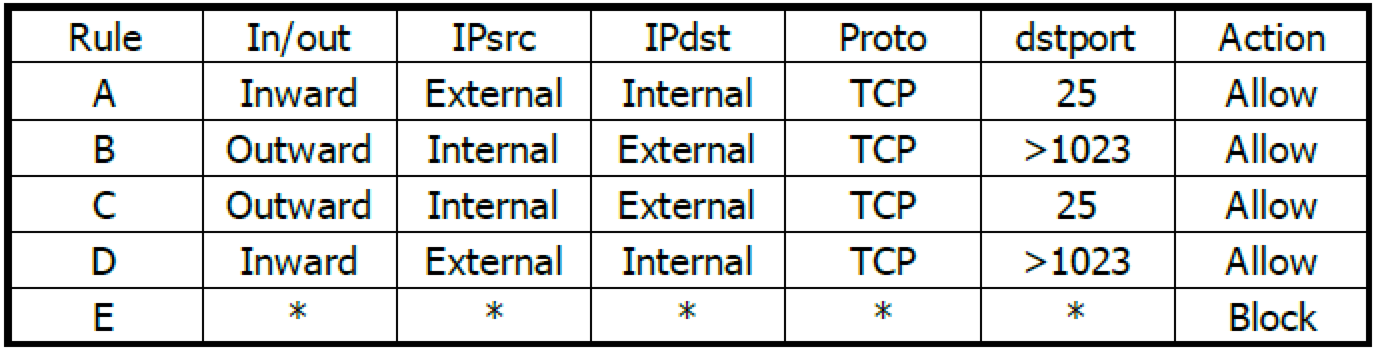
Rule	In/out	IPsrc	IPdst	Proto	dstport	Action
A	Inward	External	Internal	TCP	25	Allow
B	Outward	Internal	External	TCP	>1023	Allow
C	Outward	Internal	External	TCP	25	Allow
D	Inward	External	Internal	TCP	>1023	Allow
E	*	*	*	*	*	Block

- Scenario 2:



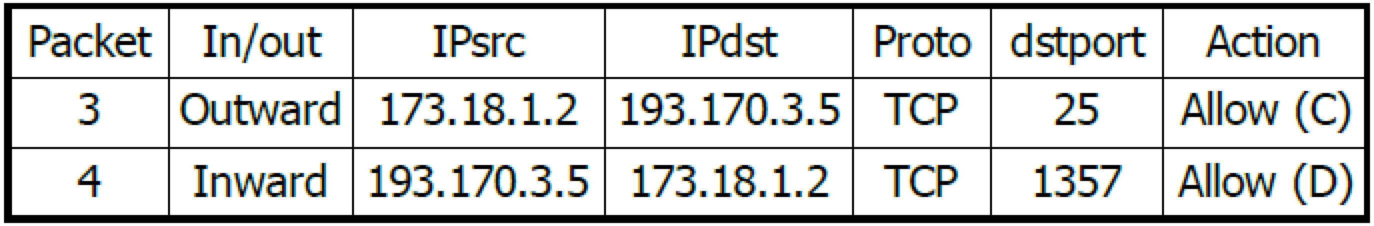
Filter rules, 4

- Example: Filtering in- and outgoing SMTP traffic. Try rules:



Rule	In/out	IPsrc	IPdst	Proto	dstport	Action
A	Inward	External	Internal	TCP	25	Allow
B	Outward	Internal	External	TCP	>1023	Allow
C	Outward	Internal	External	TCP	25	Allow
D	Inward	External	Internal	TCP	>1023	Allow
E	*	*	*	*	*	Block

- Scenario 2:



Packet	In/out	IPsrc	IPdst	Proto	dstport	Action
3	Outward	173.18.1.2	193.170.3.5	TCP	25	Allow (C)
4	Inward	193.170.3.5	173.18.1.2	TCP	1357	Allow (D)

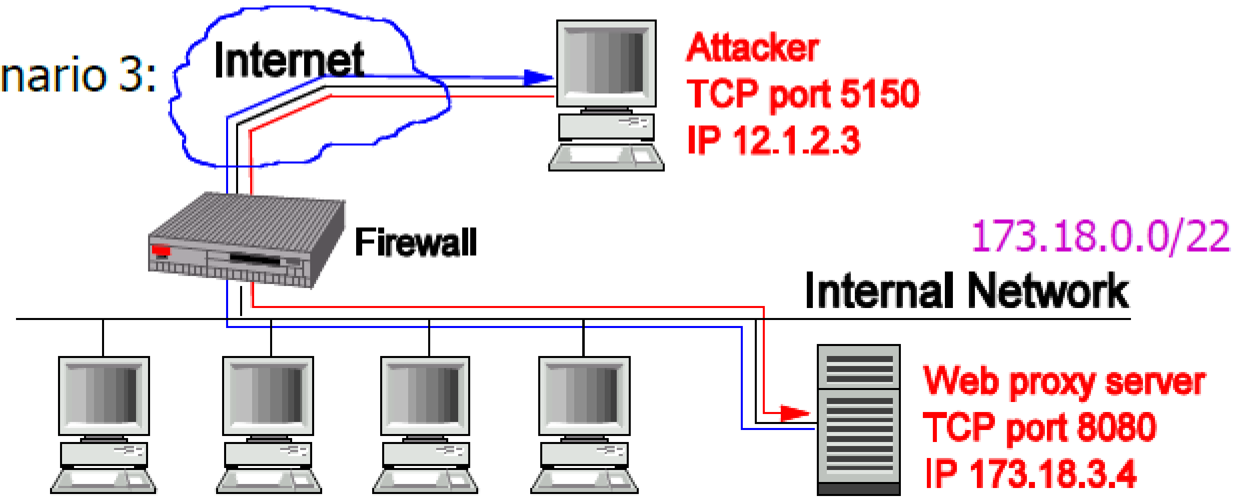
- Conclusion: This also looks OK!

Filter rules, 5

- Example: Filtering in- and outgoing SMTP traffic. Try rules:

Rule	In/out	IPsrc	IPdst	Proto	dstport	Action
A	Inward	External	Internal	TCP	25	Allow
B	Outward	Internal	External	TCP	>1023	Allow
C	Outward	Internal	External	TCP	25	Allow
D	Inward	External	Internal	TCP	>1023	Allow
E	*	*	*	*	*	Block

- Scenario 3:



Filter rules, 6

- Example: Filtering in- and outgoing SMTP traffic. Try rules:

Rule	In/out	IPsrc	IPdst	Proto	dstport	Action
A	Inward	External	Internal	TCP	25	Allow
B	Outward	Internal	External	TCP	>1023	Allow
C	Outward	Internal	External	TCP	25	Allow
D	Inward	External	Internal	TCP	>1023	Allow
E	*	*	*	*	*	Block

- Scenario 3:

Packet	In/out	IPsrc	IPdst	Proto	dstport	Action
5	Inward	12.1.2.3	173.18.3.4	TCP	8080	Allow (D)
6	Outward	173.18.3.4	12.1.2.3	TCP	5150	Allow (B)

- Conclusion: Oh, dear! That doesn't look good at all!
- Rules allow all connections where both ends use ports >1023.

Filter rules, 7

- Filtering in- and outgoing SMTP traffic.   Include srcport in rules:

Rule	In/out	Ipsrc	IPdst	Proto	**srcport**	dstport	Action
A	Inward	External	Internal	TCP	**>1023**	25	Allow
B	Outward	Internal	External	TCP	**25**	>1023	Allow
C	Outward	Internal	External	TCP	**>1023**	25	Allow
D	Inward	External	Internal	TCP	**25**	>1023	Allow
E	*	*	*	*	*		Block

- Scenario 3:

Packet	In/out	Ipsrc	IPdst	Proto	srcport	dstport	Action
5	Inw.	12.1.2.3	173.18.3.4	TCP	5150	8080	Deny (E)
6	Outw.	173.18.3.4	12.1.2.3	TCP	8080	5150	Deny (E)

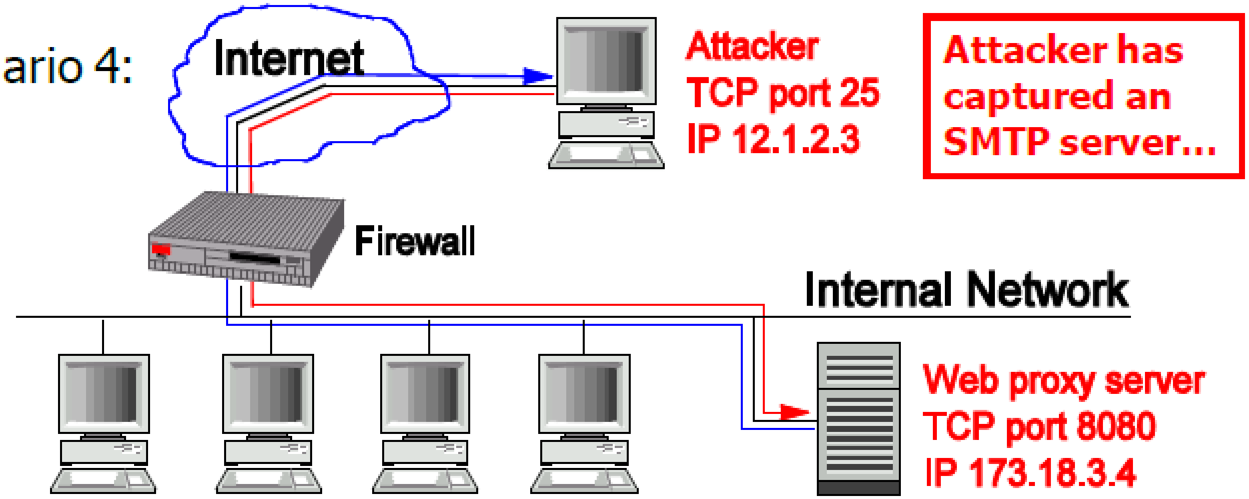
- Conclusion: This looks OK again!
- Check for yourselves that packets 1, 2, 3, 4 are treated OK.

Filter rules, 8

- Filtering in- and outgoing SMTP traffic. Include srcport in rules:

Rule	In/out	IPsrc	IPdst	Proto	srcport	dstport	Action
A	In	External	Internal	TCP	>1023	25	Allow
B	Out	Internal	External	TCP	25	>1023	Allow
C	Out	Internal	External	TCP	>1023	25	Allow
D	In	External	Internal	TCP	25	>1023	Allow
E	*	*	*	*	*		Block

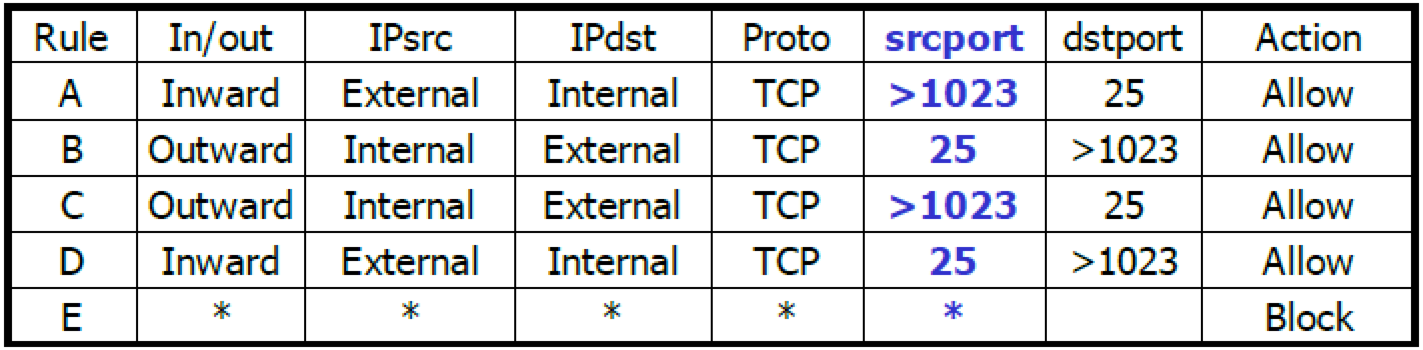
- Scenario 4:



Attacker has captured an SMTP server...

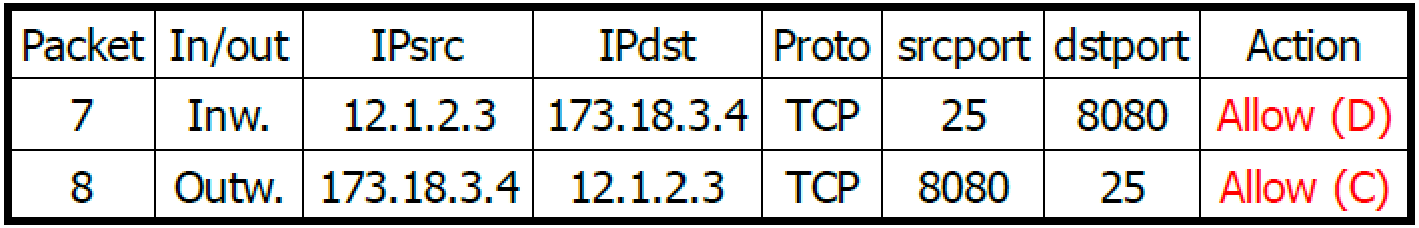
Filter rules, 9

- Filtering in- and outgoing SMTP traffic. Include srcport in rules:



Rule	In/out	IPsrc	IPdst	Proto	srcport	dstport	Action
A	Inward	External	Internal	TCP	>1023	25	Allow
B	Outward	Internal	External	TCP	25	>1023	Allow
C	Outward	Internal	External	TCP	>1023	25	Allow
D	Inward	External	Internal	TCP	25	>1023	Allow
E	*	*	*	*	*		Block

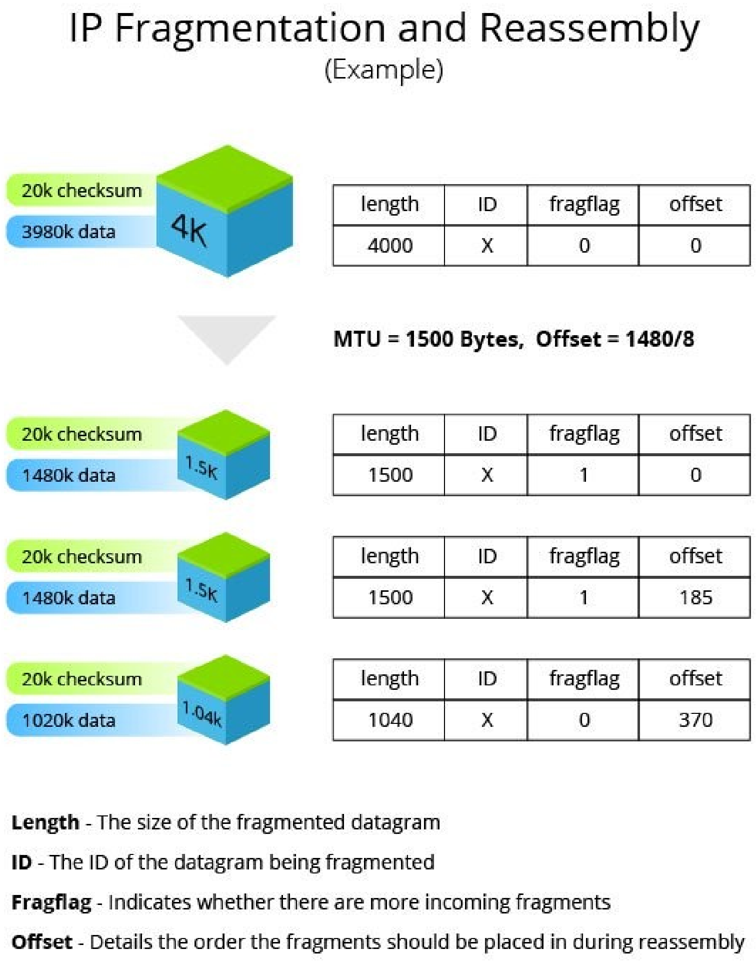
- Scenario 4:



Packet	In/out	IPsrc	IPdst	Proto	srcport	dstport	Action
7	Inw.	12.1.2.3	173.18.3.4	TCP	25	8080	Allow (D)
8	Outw.	173.18.3.4	12.1.2.3	TCP	8080	25	Allow (C)

- Conclusion: This looks bad again!
- Need yet more information (e.g. Flags) to get desired effect: Rules B and D must require ACK flag to be set in order to accept packet.

IP fragmentation



IP Fragmentation and Reassembly

(Example)

length	ID	fragflag	offset
4000	X	0	0

MTU = 1500 Bytes, Offset = 1480/8

length	ID	fragflag	offset
1500	X	1	0

length	ID	fragflag	offset
1500	X	1	185

length	ID	fragflag	offset
1040	X	0	370

Length - The size of the fragmented datagram

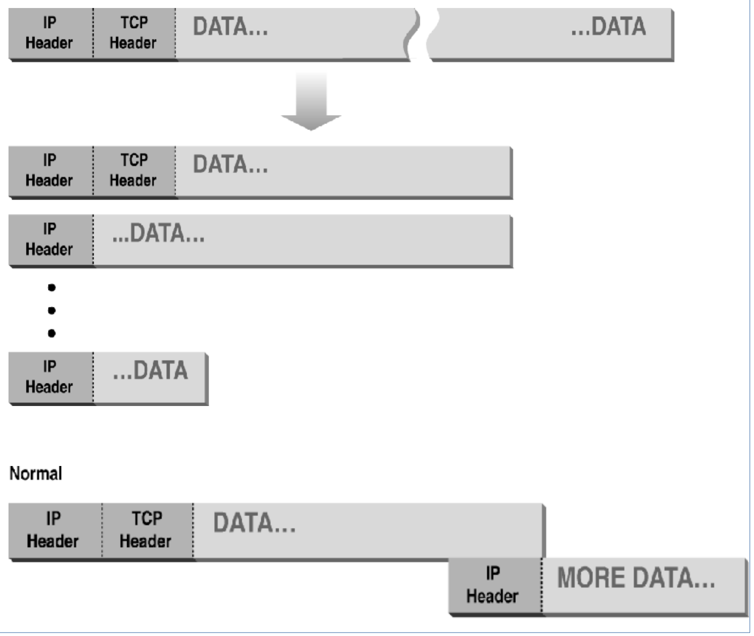
ID - The ID of the datagram being fragmented

Fragflag - Indicates whether there are more incoming fragments

Offset - Details the order the fragments should be placed in during reassembly

https://www.incapsula.com/ddos/attack-glossary/ip-fragmentation-attack-teardrop.html

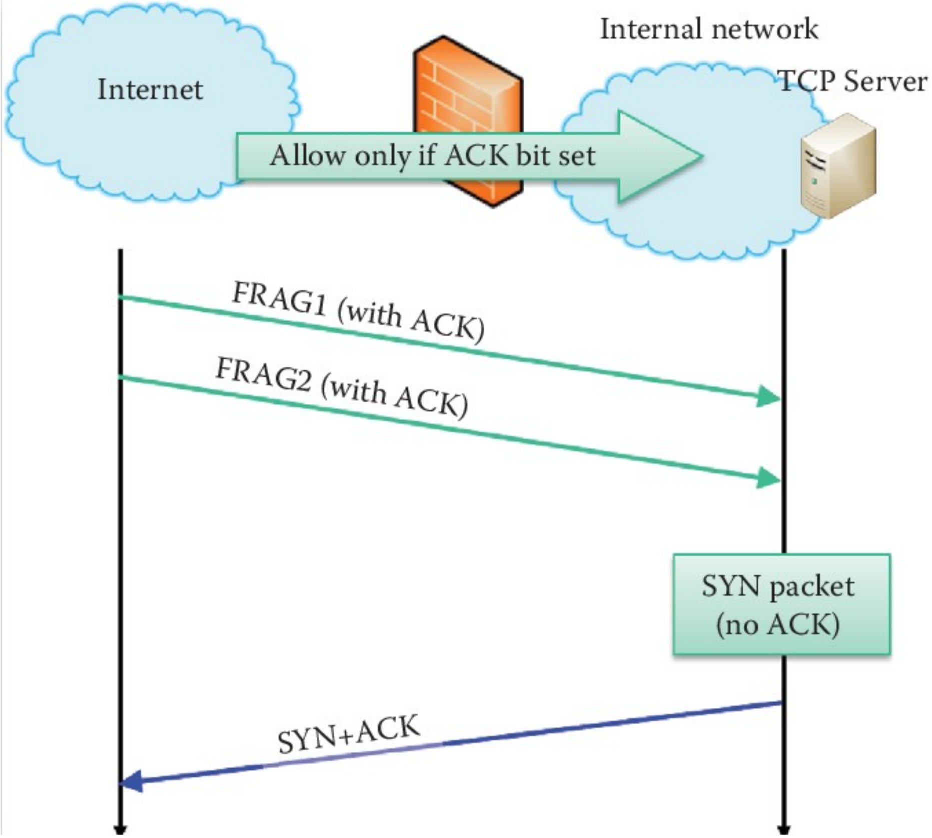
Normal fragmentation



Abnormal fragmentation

Incoming TCP connections with IP frag

- Firewall blocks any incoming TCP connection
- ACK packet is allowed for outgoing packets
- Internal host reassembles a packet with the SYN bit set because two fragment offsets are chosen in order to set the SYN bit
- Attacks
  - SYN scan
  - Create TCP connection
  - SYN flood - DoS



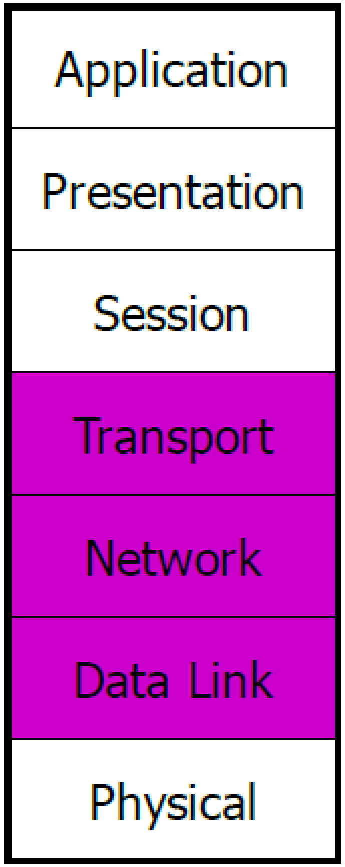
Stateful firewalls

Stateful packet inspection

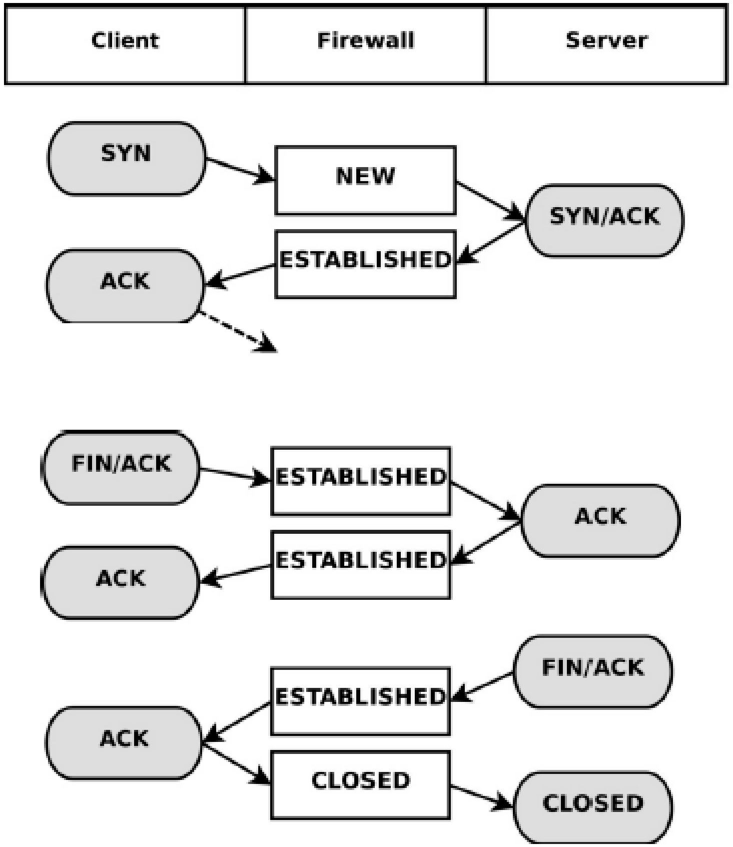
- Stateful Inspection Firewalls (or Dynamic Packet Filters) can keep track of established connections
- Can drop packets based on their source or destination IP addresses, port numbers and possibly TCP flags
  - Solve one major problem of simple packet filters, since they can check that incoming traffic for a high-numbered port is a genuine response to a previous outgoing request to set up a connection

Stateful firewall

- Considered layers


Application
Presentation
Session
Transport
Network
Data Link
Physical

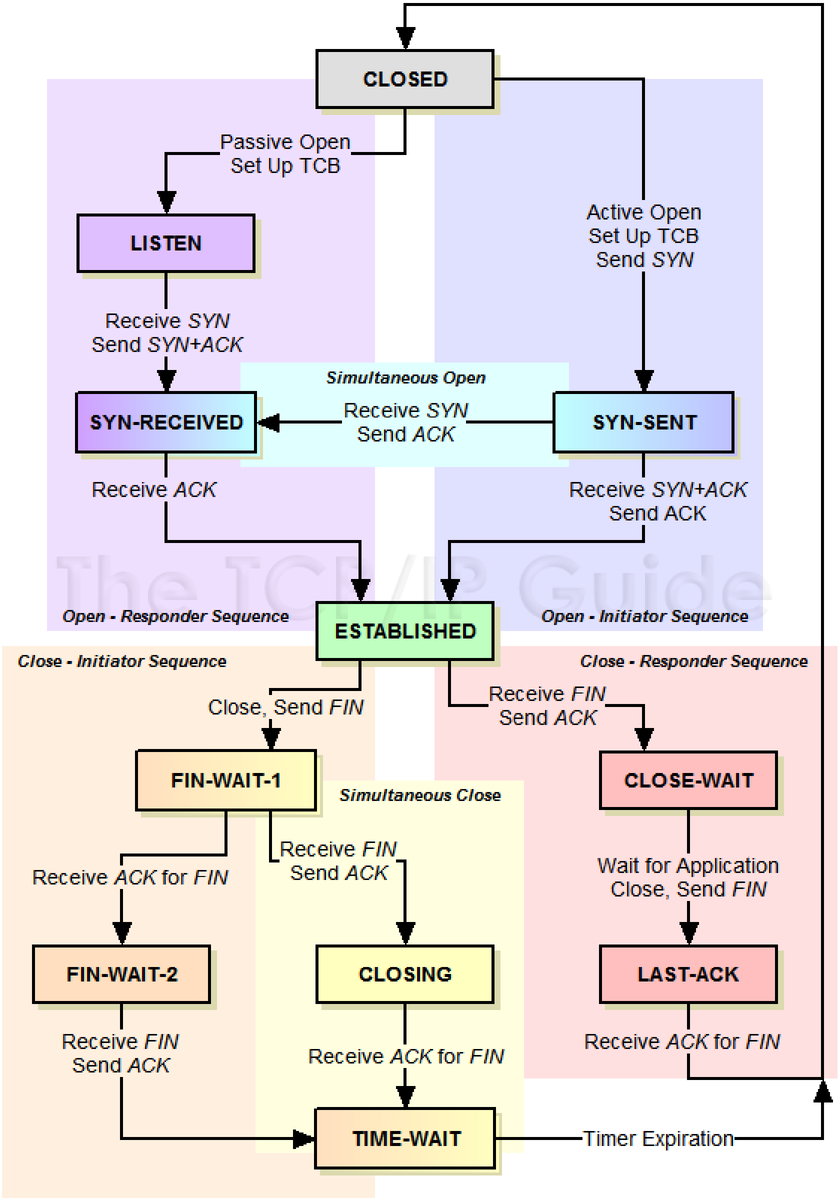
- Connection tracking



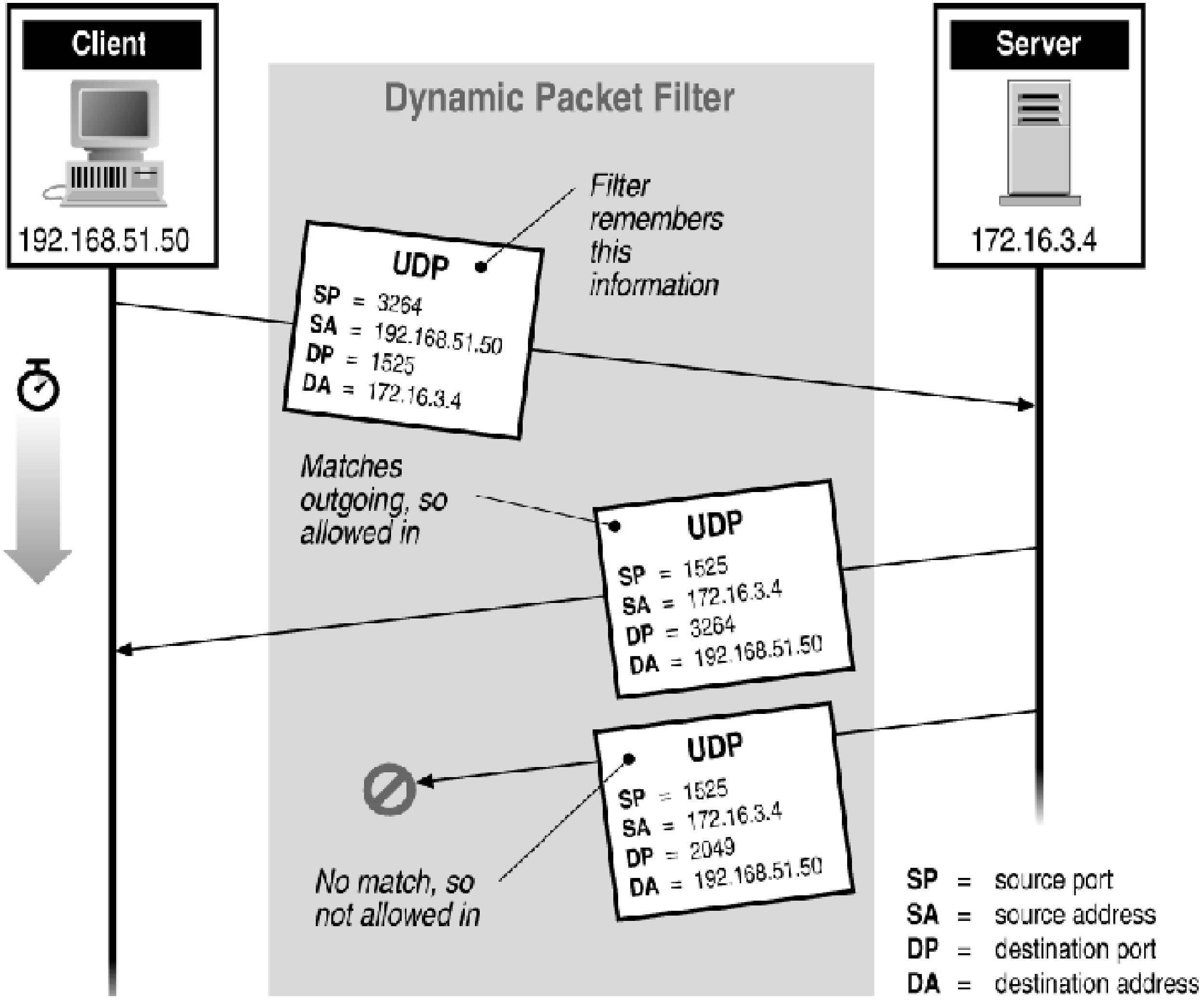
Connection tracking

Considered TCP States

- Setting up connection:
  - client calls from (high-numbered) port to port for application on server
  - server replies to (high-numbered) port on client
  - connection is considered established when the server gives correct SYN/ACK response.
- Closing connection:
  - both parties have to close the connection by sending a TCP packet with FIN flag set before connection is considered closed



Stateful firewall example

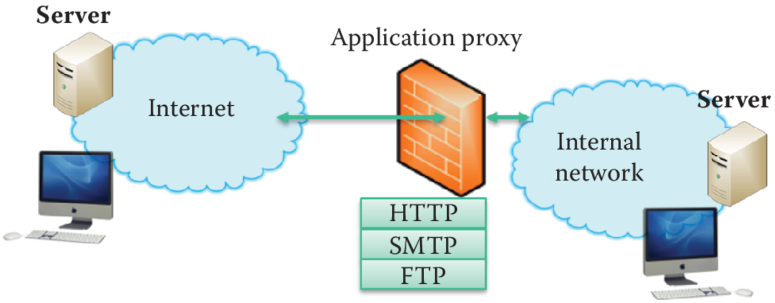


SP	=	source port
SA	=	source address
DP	=	destination port
DA	=	destination address

Other types of firewalls

Application-Level filtering (proxy)

- Deal with the details of services
- Need a separate special-purpose mechanism for each application
  - Example: mail filters, FTP, HTTP proxy
  - Big overhead, but can log and audit all activity
- Can support user-to-gateway authentication
  - Log into the proxy server with username and password
  - Example: Microsoft ISA, SQUID



Host based firewalls

- A firewall on each individual host to protect that one machine
- Selectively enable specific services and ports that will be used to send and receive traffic
  - Ex: it's unlikely that an employee would need remote SSH access to her laptop
- A host-based firewall plays a big part in:
  - reducing what is accessible to an outside attacker
  - protecting the other elements of the IT system if one of the component (ex, a process) is compromised

Proxy pro and cons

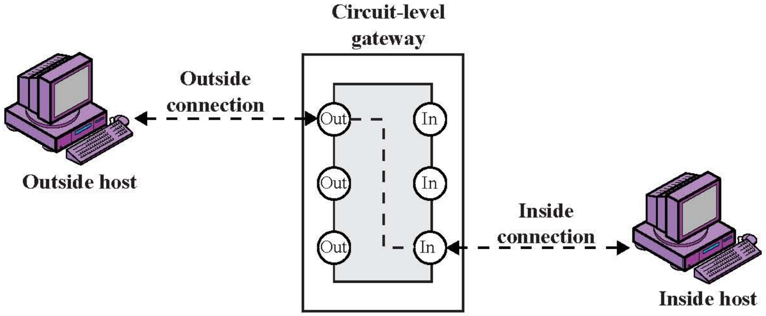
- + Logging capacity
- + Caching
- + Intelligent filtering
- + User-level authentication
- + Protect for wrong implementations
- - Introduce lag
- - Application-specific
- - Not always transparent

Circuit-level gateways (or generic proxy)

- Also known as a TCP relay
  - Able to deal with several protocols
- SOCKS (v5.0: Internet RFC1928) is the de facto standard
  - It also works with UDP
  - WinSock for Windows
- SOCKS performs at Layer 5 of the OSI model
  - The Session Layer above transport layer
    - TOR (the onion routing): socks-like interface
- The client connect to a proxy that relays its connections in a protocol-independent manner
- Provide user-authentication
- Usually no content filtering

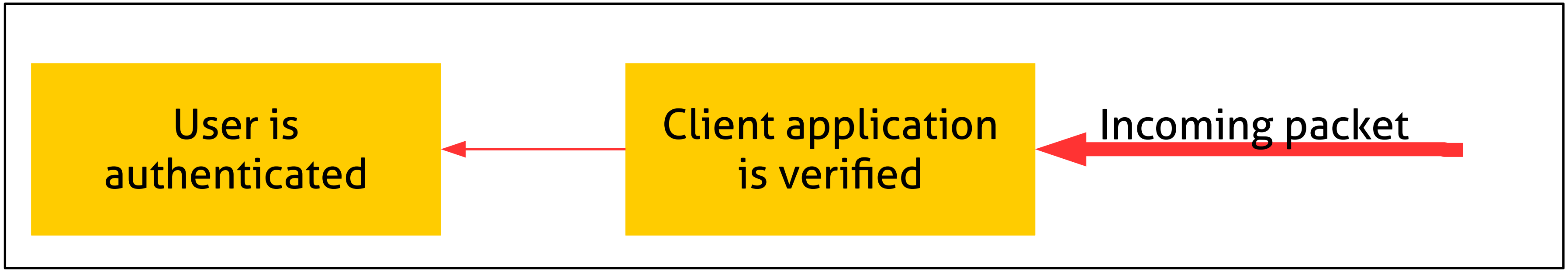
TCP relay

- Splices and relays TCP connections
  - Does not examine the contents of TCP segments
  - Can use ACL (like packet filtering, i.e. dst IP/dst port)
  - Less control than application-level gateway
- Client applications must be adapted for SOCKS
  - “Universal” interface to circuit-level gateways
- Example: `ssh -D 12345 <remote_host>`
  - More on this when talking about tunneling



Application vs circuit level gateway

Application gateway

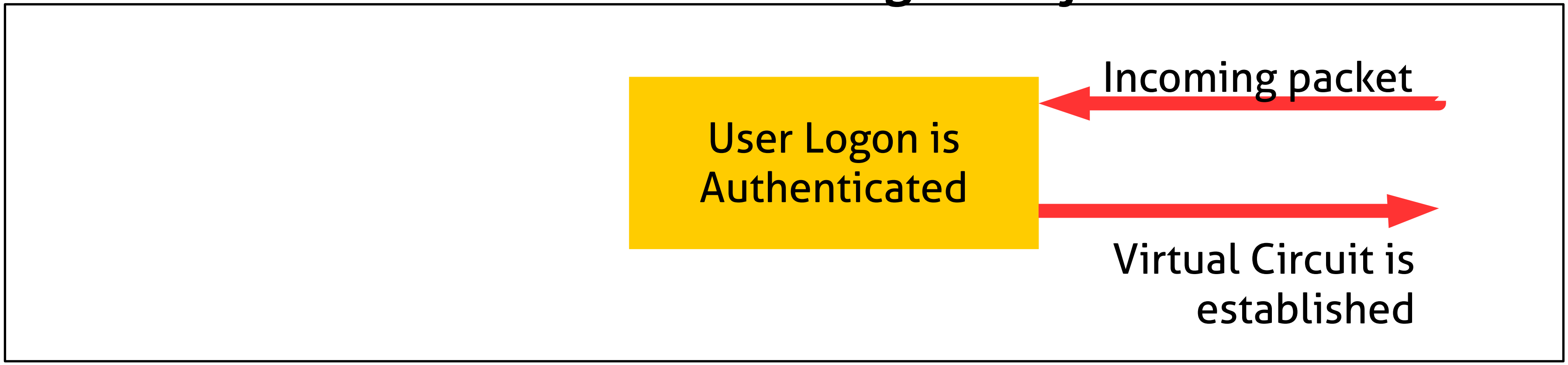


User is authenticated

Client application is verified

Incoming packet

Circuit level gateway



User Logon is Authenticated

Incoming packet

Virtual Circuit is established

Common firewall weaknesses

- No content inspection causes the problems
  - Software weakness (e.g. buffer overflow, and SQL injection exploits)
  - Protocol weakness (WEP in 802.11)
- No defense against
  - Denial of service
  - Insider attacks
- Firewall failure has to be prevented
  - Firewall cluster for redundancy

NG-Firewalls

- Next Generation firewalls try to include additional features
- Not only traffic filtering, but also:
  - Intrusion Detection System
  - VPN gateway
  - Deep Packet Inspection
  - Traffic shaping

Summary!

Summary

- Traffic regulation: routers and firewall
  - Decide the packets that can pass through the node
- Firewall architectures: where they go in the network?
  - Network segmentation and DMZ
- Types of firewalls:
  - Host firewall, stateless, stateful, application-gateway, circuit-gateway
- Stateless firewall weaknesses
  - No state, IP fragmentation

That’s all for today

- **Questions?**
- See you next lecture!
- Resources:
  - “Building internet firewalls”, Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman, O’Reilly 2nd ed. (old but still very useful for educational purposes)
    - https://docstore.mik.ua/orelly/networking_2ndEd/fire/index.htm
  - “Firewalls and Internet security: repelling the wily hacker”, William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin, Addison-Wesley 2nd ed.