Firewalls

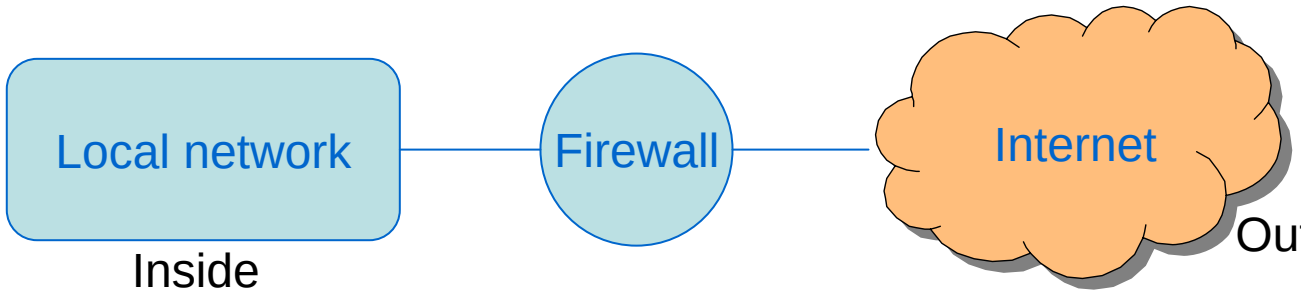
CSS 322 – Security and Cryptography

Contents

- Firewall design principles
- Packet filtering firewalls
- Application-level firewalls (proxy servers)
- Bastion hosts and firewall configurations

Protecting Computer Networks

- Internet access is no longer optional for organisations
  - Information needed by companies is available via Internet
  - End users need/want Internet for day-to-day communications
- Internet access is usually provided by one or several computers (routers) on the company network
- How do you protect the computers within the network?
  - Provide security mechanisms on all computers
    - Intrusion detection systems, anti-virus, strong cryptography
    - But maintenance is almost impossible with moderate to large sized networks (100's to 1000's of computers of many different makes)
    - Most computers have many different Internet services, which are potential security holes, available by default; and users may enable even more security holes
  - Provide a firewall that controls access between local (e.g. company) network and outside (e.g. public Internet)
    - Firewalls centrally manage access to services
    - Firewalls do what the individual computers should do (but do not)



Firewall Design Goals and Techniques

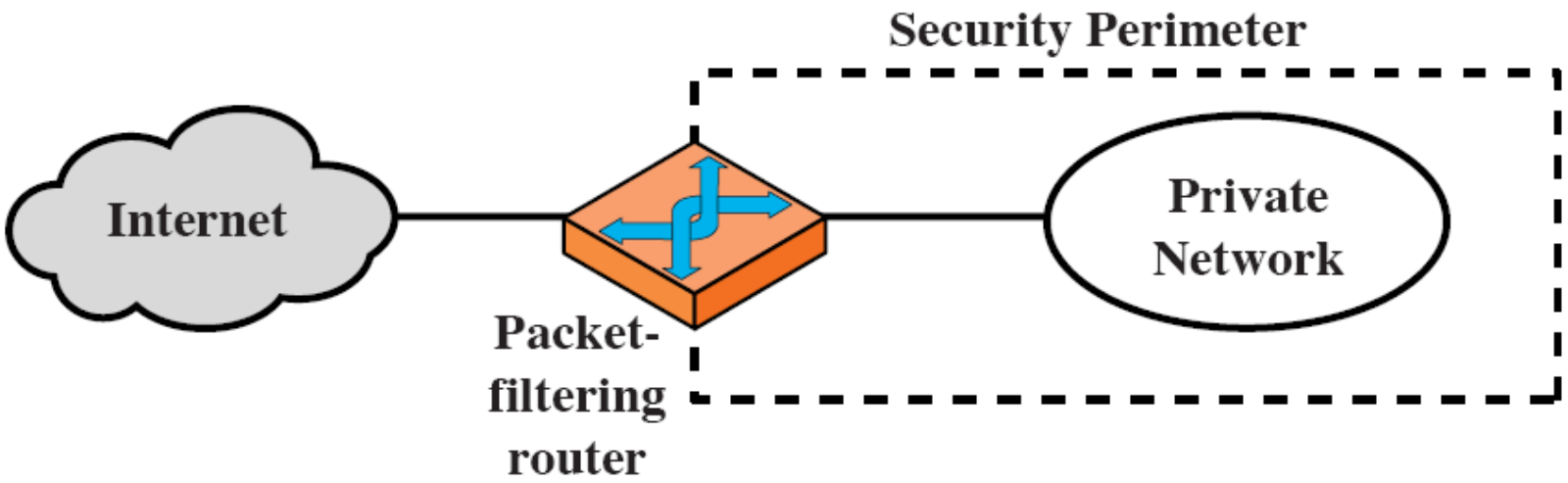
- Firewall Design Goals:
  - All traffic to/from Internet must pass through firewall
    - Physically block access to Internet, except through firewall
  - Only authorized traffic can pass through firewall
    - "Authorized" is defined by local security policy
  - Firewall must be secure
    - A "hardened" system, with trusted operating system
- General Firewall Techniques:
  - Service Control: determine what types of Internet services are allowed, often based on IP address and TCP/UDP port numbers
  - Direction Control: control the direction at which services can be accessed (e.g. request to web server outside firewall allowed; request to web server inside firewall disallowed)
  - User Control: control access based on who is the user requesting a service
  - Behaviour Control: control how particular services are used (e.g. firewall may filter email/spam)

Firewall Capabilities and Limitations

- Firewall Capabilities (what it should do):
  - Single point of control: keeps unauthorised users out of internal network; prevents access to vulnerable services
  - Monitor security-related events
  - Useful location for non-security-related Internet functions: Network Address Translation and usage monitoring
  - Can act as IPsec tunnel end-point (e.g. for VPNs)
- Firewall Limitations (what it cannot do):
  - Protect against attacks that bypass firewall
    - E.g. Internal computer connects to Internet via dial-up and ISP
  - Protect against internal threats
    - E.g. Employee inside the network attacking another internal computer
- Weakest Link Principle:
  - Security of your system is only as effective as the weakest link
  - If multiple points of Internet access, each point needs a firewall, and all the firewalls should be configured identically

Packet Filtering Firewalls

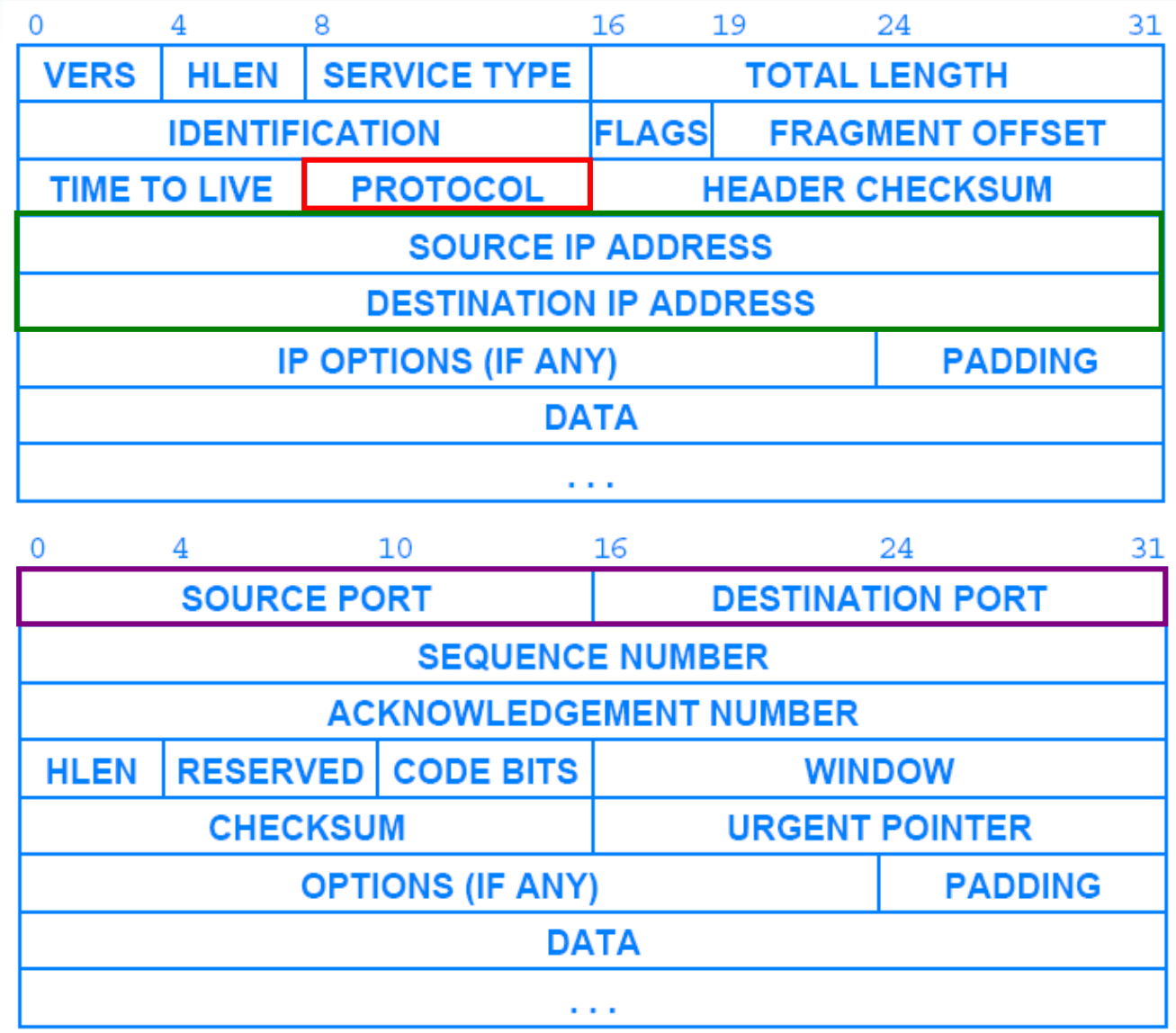
- A router that applies set of rules to each incoming and outgoing IP packet
  - Incoming: Internet to local network
  - Outgoing: Local network to Internet
- Why a router?
  - A router normally receives IP packets, looks up destination address, and forwards the IP packet
  - Firewall on router: also look at details of IP packet before deciding whether to forward (accept) or not (reject/discard)



Packet Filtering Rules

- Packet filtering router looks at following information:
  - Source and destination IP addresses: IP addresses are carried in the IP packet header
  - Source and destination port numbers: port numbers are carried in the transport (TCP/UDP) packet headers
  - Transport protocol identifier: IP packet includes a field to specify if TCP or UDP (or another protocol) is being used
  - Interface of router: if more than two router interfaces, then firewall may check where packet came from, and where is it destined to
- Packet filter rules:
  - Set of rules that determine if packet should be accepted (forwarded) or rejected (dropped)
  - Rules usually processed in order
  - Need a default policy if a packet does not match any rule
    - Best practice is to “reject” all packets, except if you specifically allow it via a rule
      - This may cause problems for some users – their traffic may be rejected until firewall administrator explicitly creates a rule for it
- No standard way to describe firewall packet filters
  - Products implement filters in different ways

IP and TCP Headers



Packet Filter Rules: Example 1

ARRIVES ON INTERFACE	IP SOURCE	IP DEST.	PROTOCOL	SOURCE PORT	DEST. PORT
2	*	*	TCP	*	21
2	*	*	TCP	*	23
1	128.5.0.0 / 16	*	TCP	*	25
2	*	*	UDP	*	43
2	*	*	UDP	*	69
2	*	*	TCP	*	79

- Table specifies packets to be *dropped*
- Set of rules specify:
  - Block all packets destined to following services on internal network:
    - FTP (port 21); TELNET (23); WHOIS (UDP port 43); TFTP (69); FINGER (79)
  - Block all packets coming from internal network 128.5.0.0 (subnet mask 255.255.0.0) and destined to external email server (port 25)

Packet Filter Rules: Example 2

- Following tables specify action to be taken; by default the action is discard/block the packet

inside outside

	action	ourhost	port	theirhost	port	comment
A	block	*	*	SPIGOT	*	we don't trust these people
	allow	OUR-GW	25	*	*	connection to our SMTP port

A. Inbound email is allowed, but only to the gateway host (OUR-GW); External host SIPGOT is not allowed to send any packets

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

B. Explicit statement of the default drop policy (normally this rule does not have to be specified; it is assumed by default)

Packet Filter Rules: Example 2

C

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

C. Aim: any internal host can send email to outside
- Responses that come from source port 25 are accepted
- Problem: A malicious (external) host (not using SMTP) could access internal machines by setting source port to TCP

source destination

D

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

D. Aim: fix the problem of part C
- Only allow selected hosts on internal network to send
- Only allow TCP ACKs from other computers with source port 25
- (Not the difference in the tables)

Packet Filter Rules: Example 2

E. Handling FTP connections

- FTP sets up two TCP connections:
  - Control connection, for sending requests for files etc (e.g. GET file)
  - Data connection, for transferring the files
  - Control uses port 21, but data uses dynamic port
- Servers uses well-known ports from 0 to 1023; ports higher than 1024 are for other connections
- Rule 1 allows our internal hosts to initiate connections to any server (e.g. initiate FTP control connection)
- Rule 2 allows replies to our hosts to be sent (e.g. respond to FTP control connection)
- Rule 3 allows external hosts to send traffic to a (non-server) port on internal machines (e.g. FTP data connection)

E

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	*		our outgoing calls
allow	*	*	*	*	ACK	replies to our calls
allow	*	*	*	>1024		traffic to nonservers

Issues with Packet Filter Firewalls

- Advantages:
  - Simple; transparent to users; very fast
- Disadvantages:
  - As they do not inspect upper layer protocols (e.g. applications):
    - Cannot detect application-specific attacks
      - E.g. cannot detect a malicious HTTP GET request
    - Can only log IP traffic information (cannot obtain detailed statistics for each application)
  - Vulnerable to address spoofing attacks
    - Attacker changes the source IP address to be something different than it actually is
    - As there are only several (5 or 6) variables in the rules, often rules become quite complex, meaning susceptible to mistakes (and hence security breaches)

Attacks and Countermeasures

- IP Address Spoofing
  - Attacker (from outside) transmits packet with source IP address changed to IP address from inside
    - Aim is for firewall to allow packet, as it passes a rule for packets from inside the network
    - Easy to stop: firewall drops all packets with internal IP address, but arriving on external interface
- Source Routing Attacks
  - IP allows a packet to include a source route, that is the route the packet should take across the Internet (seldom used option)
  - Attacker sends a IP packet with source routing, with aim that firewall will not investigate the source routing details
    - Easy to stop: firewall drops all packets that use source routing
- Tiny Fragment Attacks
  - If an IP fragment is received, normally a firewall will filter based on first fragment and block/allow all subsequent fragments
  - Attacker creates very small IP fragments, so TCP header information is spread across multiple fragments
    - Aim is for firewall to examine first fragment, and since not enough information, allow that fragment, and all following fragments
    - Stop by enforcing rule that requires the first fragment to contain a pre-defined amount of TCP header, so firewall can make correct decision about block/allow

Stateful Packet Inspection

- Client/server applications use well-known port for server and dynamic port for client
    - Ports 0 to 1023 for well-known servers
    - Ports 1024 to 49151 for registered services
    - Ports 49152 65535 for dynamic assignment
  - Firewall can allow/block messages from internal clients to external servers via destination port number
  - A simple packet filter firewall will allow all responses to any client port number (e.g. greater than 1023 or 49151)
    - But these may leave vulnerabilities (holes) available for attacker to use
- Stateful packet inspection
  - Firewall maintains details about each TCP connection initiated
    - For each connection, firewall stores: source client IP/port; destination server IP/port; TCP connection state
    - Firewall can remove state after timeout or by monitoring TCP packets to determine when connection finishes (e.g. TCP FIN, RESET)
  - Firewall will only accept responses to ports on internal hosts that it has record of
  - Requires extra overhead of maintaining connection information

SPI Example

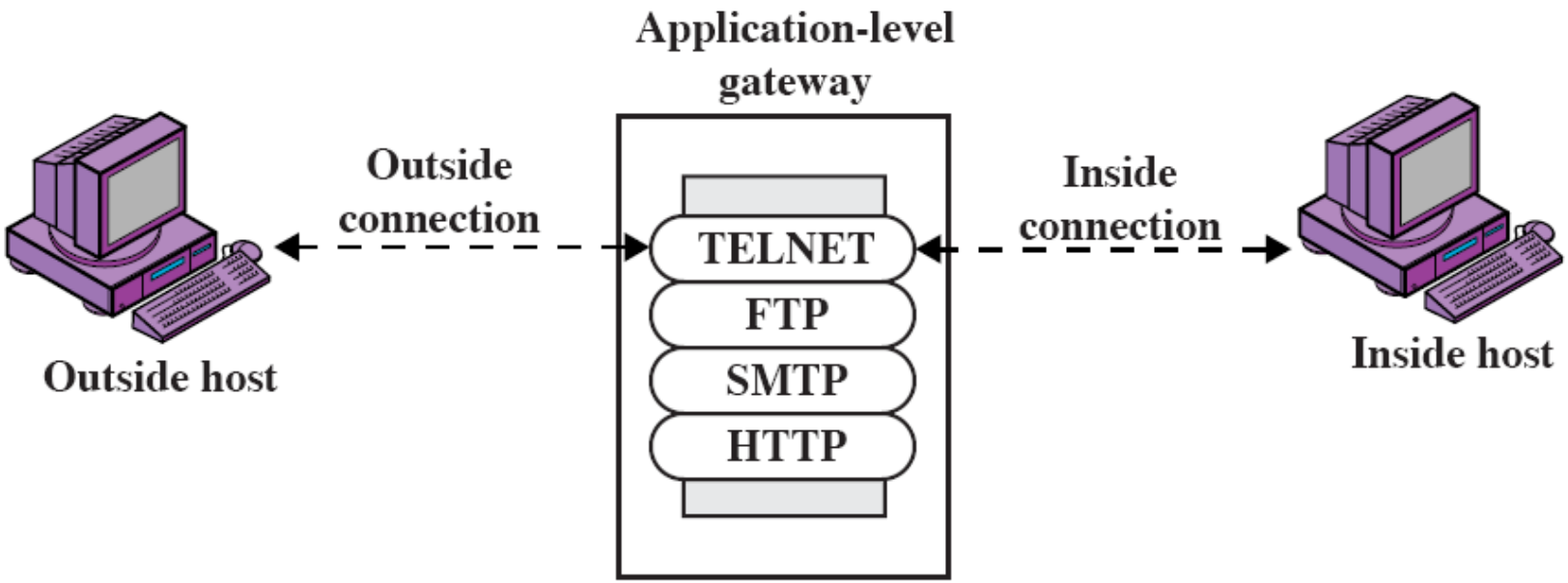
Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.9.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	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

A TCP packet from 210.9.88.29, port 80 will be accepted if the destination is 192.168.1.100, port 1030

But if the destination is 192.168.1.100, port 1031, the packet will be dropped

Application-level Firewall

- Application-level gateway (or proxy server) receives connection requests from internal clients
  - Proxy usually only allows certain applications (e.g. email, web, but may prohibit remote login or FTP)
  - Proxy inspects the requests and forward them on to external server
- (A modified version, e.g. circuit-level gateway, not only intercepts requests but also creates a new TCP connection – so in fact there



Issues with Application-Level Firewalls

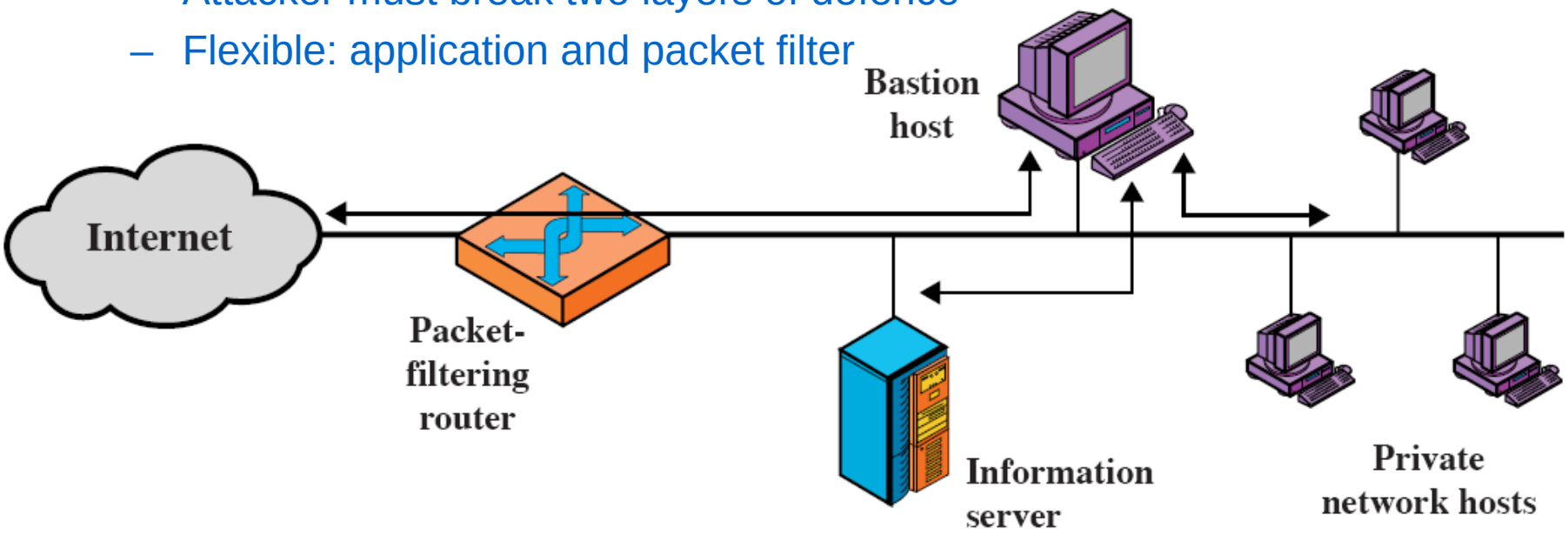
- Advantages:
  - Generally more secure than packet filter firewalls
    - Can inspect application specific data, including content
    - Can also be used to perform virus detection, spam filtering
  - Provide detailed application-level logs
  - Can be used as content cache: content that has recently been requested can be cached for future access
  - IP addresses of individual internal clients are hidden; external nodes can only see the proxy IP address
- Disadvantages:
  - Additional processing is needed by firewall
    - Proxy/gateway must examine details of all packets, as well as understand format of many protocols

Bastion Host

- A strong/secure host in the network used as application-level firewall
  - Usually executes a secure operating system
  - Only essential services are installed, including proxy applications for HTTP, email
  - May require internal uses to provide additional authentication to access proxy services
  - Proxy applications are usually very secure implementations of protocols used
    - Remove insecure commands, small amount of code
    - Easy to check for security flaws
  - Each proxy is independent of other proxies, and runs in its own secure space on the file system
- Used in various firewall configurations …
  - Screened host firewall (single homed, dual homed bastion host)
  - Screened subnet firewall system

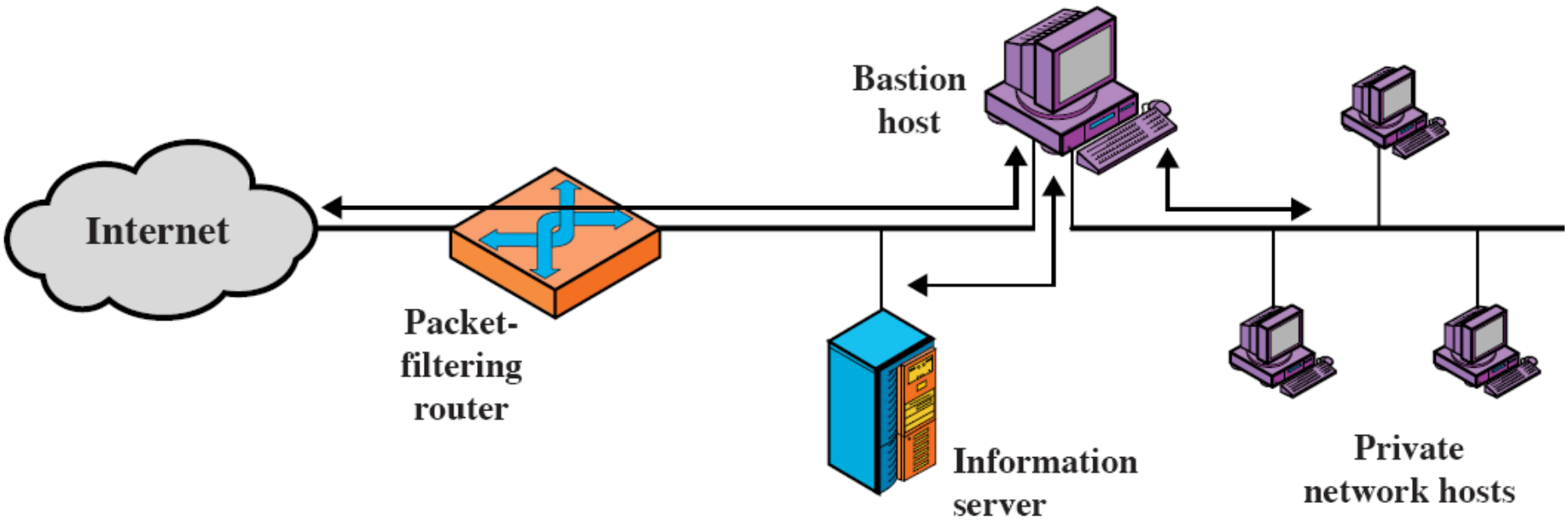
Single Home Bastion Host

- Packet-filtering firewall/router and bastion host
- Firewall configured so only:
  - Accept packets from external network only to bastion host
  - Allow out (to external network) packets from bastion host
- More secure than packet-filter firewall or application-level firewall on own:
  - Attacker must break two layers of defence
  - Flexible: application and packet filter



Dual Homed Bastion Host

- Single-homed bastion host is vulnerable of router (packet filter firewall) is compromised
- Dual homed host physically forces all traffic through bastion host – must compromise both systems to get through
- (Information server can provide be a web server serving documents to the Internet)



Screened Subnet Firewall System

- Most secure of three approaches
  - Two routers/firewalls used; bastion host sits on a separate subnet than private network
    - Called a DeMilitarised Zone (DMZ)
  - Internet (external) hosts can access DMZ; private (internal) hosts can access DMZ
    - But traffic does not flow across DMZ
  - Internet hosts do not know about structure of private network