Network Security

AA 2015/2016

System hardening

(Application Firewalls)

Dr. Luca Allodi

Some slides from M. Cremonini

Stateful Packet Filtering

- Called *Stateful Inspection* or *Dynamic Packet Filtering*
- Maintains a history of *previously seen packets* to make better decisions about current and future packets
  - Connection state maintained in a connection table
- Define rules to open state
- It's possible to use existent state to control future packets
  - e.g. explicit rule for TCP SYN in LISTEN state
    - "NEW" connection in IPTABLES
    - Subsequent packets can be filtered using the connection table
      - E.g. allow any packet for an ESTABLISHED connection

Pseudo-states

- Stateful firewalls allow user to define states over stateless protocols
  - e.g. UDP traffic is stateless → use <sip,sport,dip,dport> to correlate traffic
- For these protocols there is no termination sequence
  - e.g. TCP's FIN 4-way handshake
  - Typically set a time-out wherein pseudo-state is defined
- Traffic of stateless protocols depend on application, not on protocol itself
  - May be hard to manage, application-specific

Stateful firewall rule example

- Possible states (iptables with conntrack)
  - NEW → packet trying to open a not-yet existent connection
  - ESTABLISHED → incoming packet is relative to a connection already initiated
  - RELATED → packets that are stating a NEW connection but related existing one (needed by some applications – e.g. FTP)
  - INVALID → none of the above → e.g. incoming packet with ACK but not belonging to ESTABLISHED connection → can you filter this with static filtering?

- Say you want to prevent ACK scans
  - Stateful rule:
    ```
    iptables –A INPUT –i eth0 –m state –state
    INVALID – j DROP
    ```
  - Static rule → will this be a good rule?
    ```
    iptables –A INPUT –i eth0 -p tcp
            --tcp-flags ACK – j DROP
    ```

Another example

- Example rule: allow all incoming traffic related to an existing connection

```
iptables –A INPUT –i eth0 –m state --state
     ESTABLISHED,RELATED  -j ACCEPT
```

- Mixed rules also possible:

```
iptables -P INPUT DROP
iptables -A INPUT -i ! eth1 -j ACCEPT
iptables -A INPUT -m state –state
     ESTABLISHED,RELATED -j ACCEPT
```

Application firewalls

- Staefulness consider also application layer
  - "Deep packet inspection"
  - Can keep track of and deny others
  - e.g. FTP PORT command
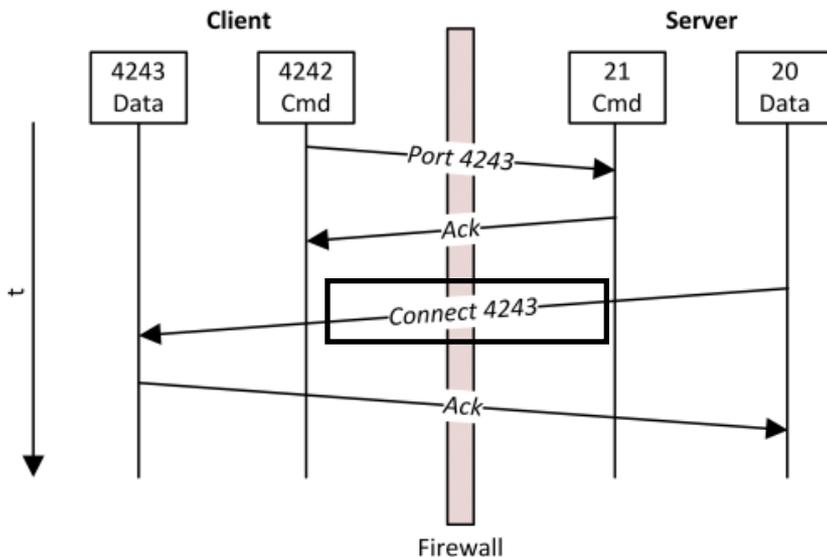
- FTP commands are passed to port 21
- In "Active mode" the server opens a connection with the client, and chooses dport
  - this happens with PORT command
- Application firewall can detect PORT command and act on packet
 - Simple stateful firewall can not easily manage this

Stateful and app firewalls: pros and cons

- Pros
  - Allow user to express more powerful rules
  - Policy definition is much simpler than with static packet filtering
  - Very diffused in all modern firewalls

- Cons
  - Severe impact on firewall performance
  - Deep packet inspection significantly slows down packet check
  - Application support may be very complicated
    - Typically provided as "modules"

Proxy

- A network component that mediates network communications
- Untangles the otherwise direct communication between client and server
- Proxy acts both as a server (that receives remote connection) and as a client (that forwards the connection to its real destination).

These IPs depend on conf either private or public

Does not create actual connection with Client A

Client A

dIP: x

Internet

x

http

Webserver

k

Proxy

y

Client B

dIP: x

May or may not know that X is proxy and not final destination

FTP server
z
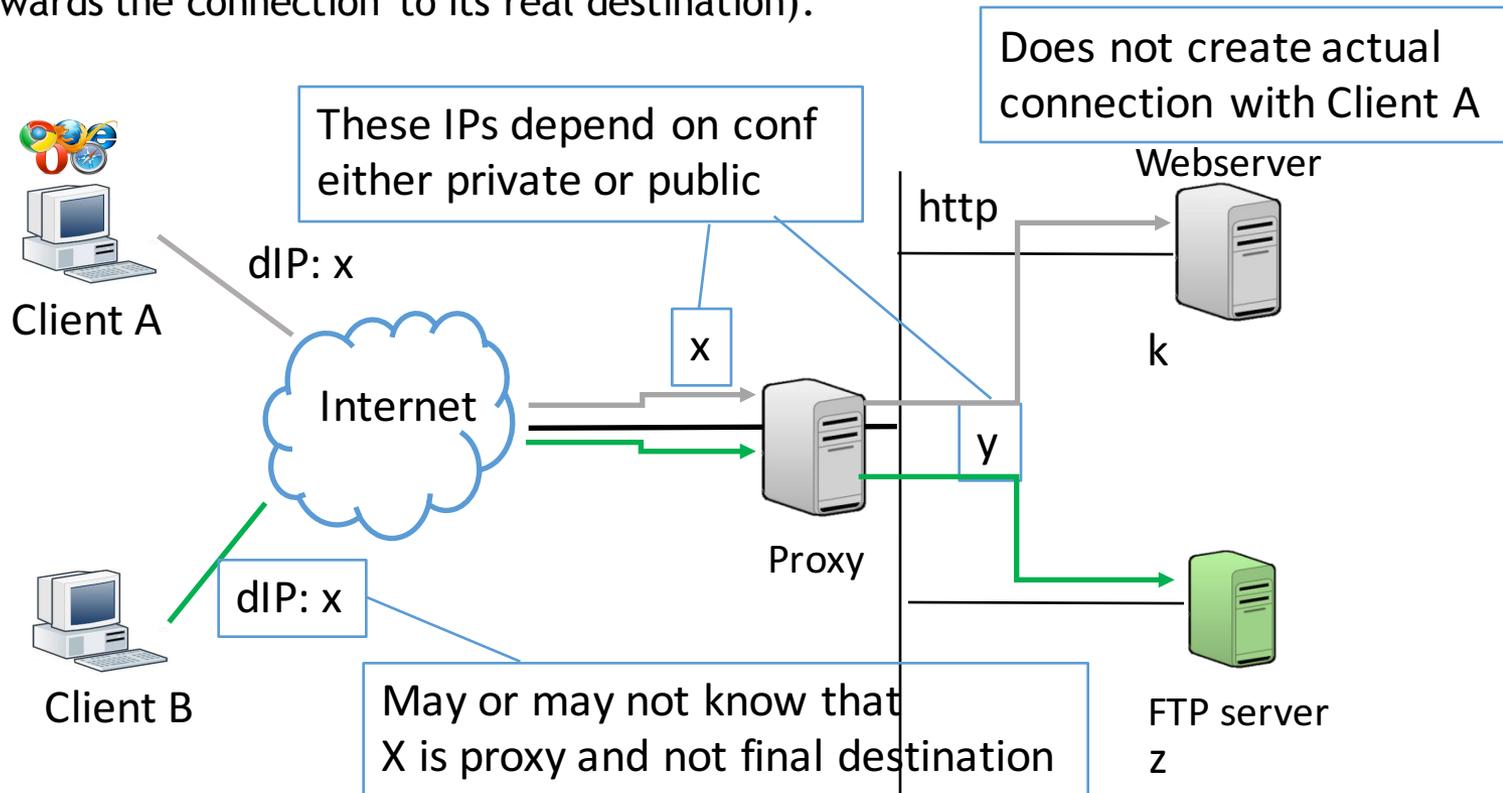
Open proxy

- Proxy connects any client on the internet to any server on the internet
- Clients knows real destination of packet
- Server can not normally know by whom was the packet originated

Client A

1.1.1.1

Internet

Public IPs

1.1.1.1

1.1.1.2

http

Webserver

1.1.1.3

Open Proxy

Client B

1.1.1.1

Tell 1.1.1.1 to forward packet to 1.1.1.4:21

FTP server
1.1.1.4
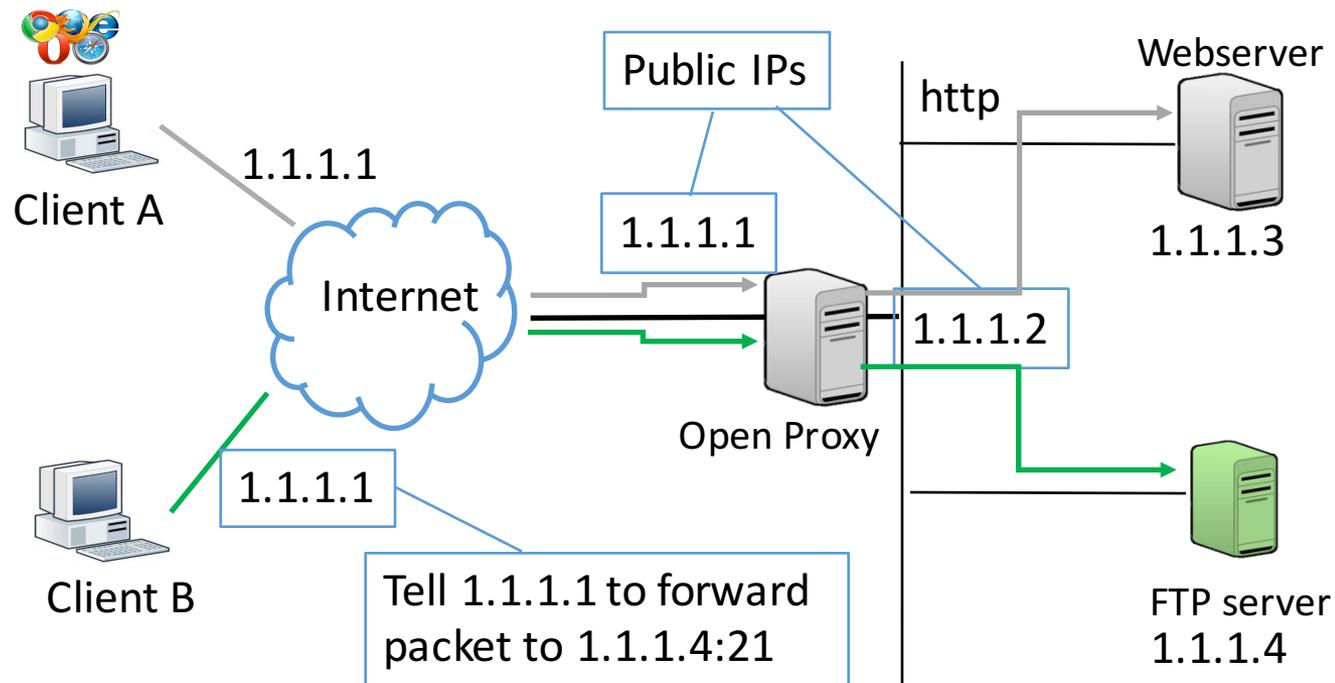
Open proxy - characteristics

- Enables the user to achieve some level of anonymity on the network
  - *Anonymous proxies*
  - Server should not be able to collect source IP
    - Some techniques exist to overcome this
    - Force the client to communicate its IP through third party services or plugins (e.g. flash)
- Trust issues → all trust is put on proxy service
  - This may or may not be sufficient depending on application
  - OK to bypass organisation's blacklist (e.g. block facebook.com)
  - Probably not trustworthy for more sensible Internet traffic
    - Confidential/secretive/illegal exchange of information
  - May be used as a malware distribution server
    - Malicious proxy embeds malware in response packet

Reverse Proxy

- Mediates connection between Internet clients and servers on an internal network it protects
- Can embed firewalling capabilities; may sit on border router.
- Client talks directly to Proxy; Proxy forward to internal servers; neither internal servers or clients know real origin/destination of packet.

Client A

1.1.1.1:80

Internet

Client B

1.1.1.1:21

Company's perimeter

1.1.1.1

Rev. Proxy

192.168.1.1

http

Webserver

192.168.1.2

FTP server
192.168.1.3
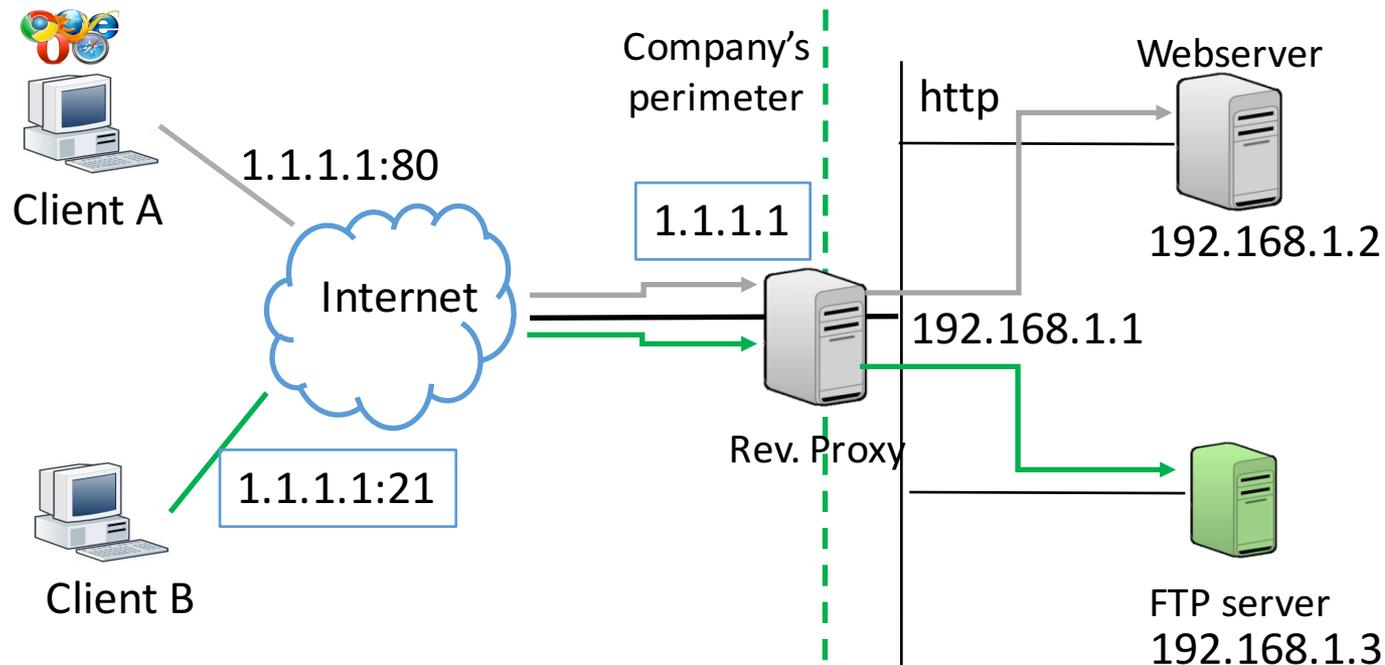
Reverse proxy - characteristics

- May hide properties of internal servers
  - IPs, non-custom service ports, versioning
    - If too aggressive may cause disservices
      - e.g. declares fake server version that breaks the protocol

- May be used for load balancing
  - Several internal replicas of a webserver
  - Proxy automatically balances the load by forwarding client's connection to most appropriate internal server
    - e.g. least busy server gets the connection
  - May be used to cache server's content → answer directly to requests for which a cache entry exists

Application Level Proxy

- Also called application proxy
- Acts as a relay of application-level traffic
- All connections are mediated by the GW
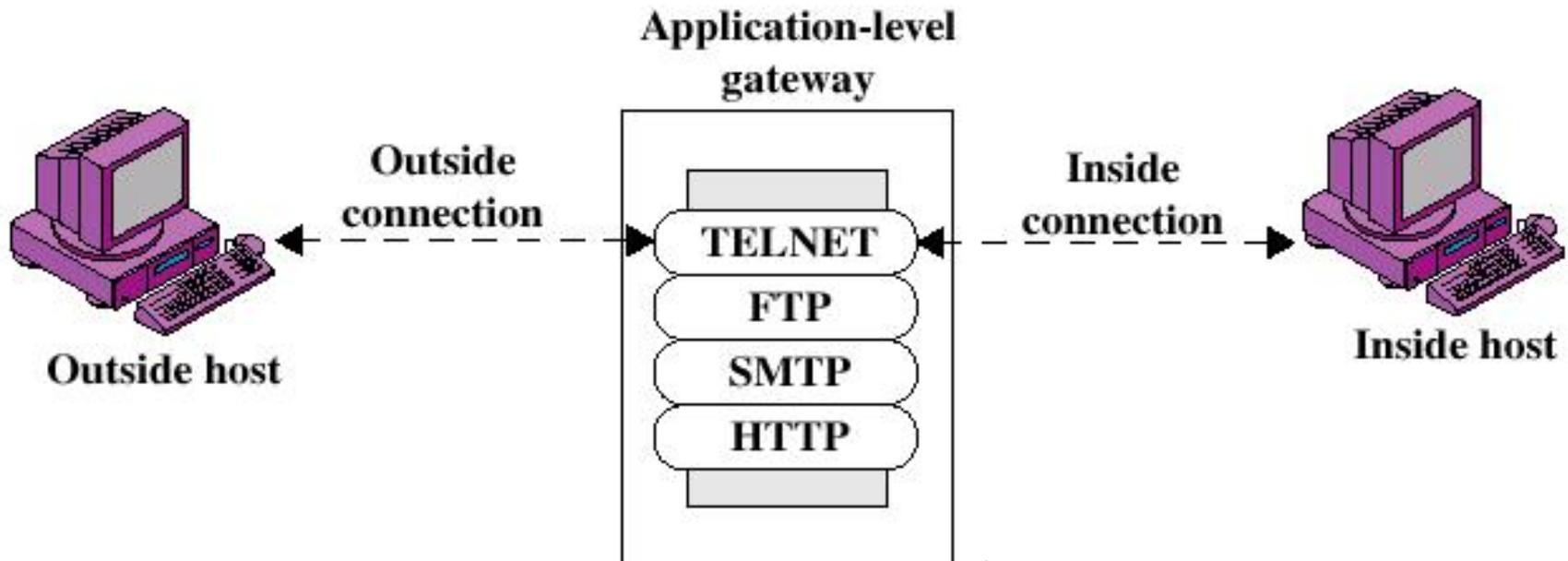
Application Gateway: Pros

- Advantages: by <u>not</u> permitting application traffic directly to internal hosts
  - *Information hiding*: names of internal systems are not known to outside systems
  - Can limit capabilities within an application
  - *Robust authentication and logging*: application traffic can be pre-authenticated before reaching host and can be logged
  - *Cost effective*: third-party software and hardware for authentication and logging only on gateway
  - *Less-complex filtering rules for packet filtering routers*; easier stateful firewall implementations
  - More secure
- Cons
  - Keeping up with new applications
  - May need to modify application client/protocols
    - Custom implementation may be expensive

Circuit-level Gateway

- Also called circuit-level proxy
- Usual, when there is a trust to internal users
- No firewalling capabilities → simply crosses client connection to inside host
  - The gateway typically relays TCP segments from one connection to the other without examining the content
  - Operates at L4 on OSI scale

Circuit-level gateway

Outside host — Outside connection — Out / In / Out / In / Out / In — Inside connection — Inside host
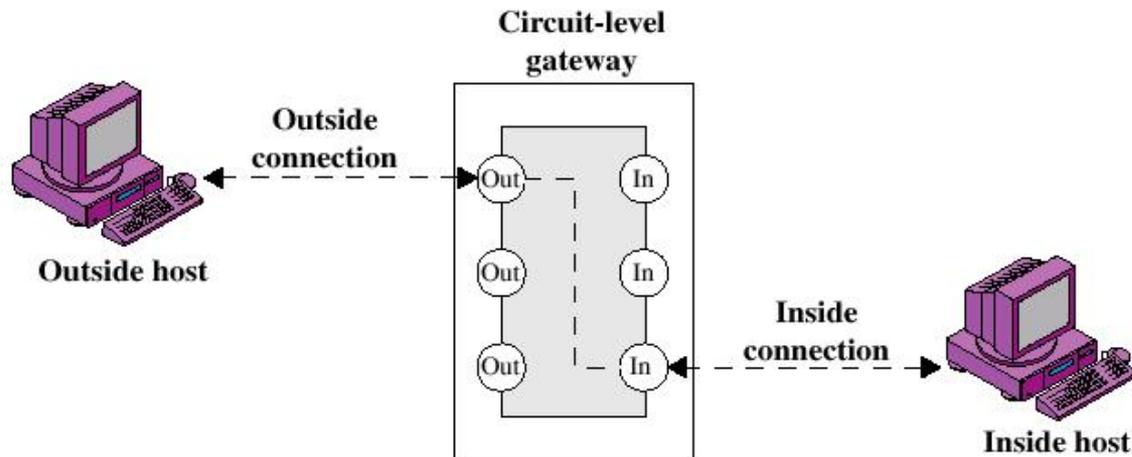
Network Address Translation

- Application gateways operate at level 7 on the OSI scale (application layer)
  - Powerful application and traffic control
  - Slow and application-dependent
- NAT operates at level 3 (network layer)
  - Acts as a L3 reverse proxy
  - Maps <sourceip,dport> to <destinationip, dport>
    - Stateful connection table keeps track of matching
    - Port Address Translation (PAT) used to resolve conflicts
      - E.g. two incoming and independent TCP connection with same source port → NAT translation must assign different sports and correctly map connection back to source IPs

Firewall Basing

- Stand-alone machine running common OS (Unix, Windows)
- Software module in router or LAN switch
- Bastion host
- Host-based firewall
- Personal firewall

Bastion Host

- A system identified by the firewall administrator as a critical strong point in the network´s security
- The bastion host serves as a platform for an application-level or circuit-level gateway
- **Characteristics:**
  - Executes on a secure version of the OS (hardened system)
  - Only essential services
  - May require additional user authentication before accessing proxy services; each proxy service may require also its own
  - Each proxy maintains detailed audit information
  - Each proxy is small software package suitable for verification
  - Each proxy is independent
  - Each proxy runs as a non-privileged separate user

Host-based Firewall

- Software module used to secure an individual host
- Available in many operating systems
- Common location for such firewalls is a server
- **Advantages**
  - Filtering rules can be tailored to the host environment (specific rules for the servers)
  - Protection is provided independent of topology. Thus both internal and external attacks must pass through the firewall
  - In conjunction with stand-alone firewalls, the host-based firewall provides an additional layer of protection

Personal Firewall

- Personal firewall controls the traffic between a personal computer or workstation on one side and the Internet or enterprise network on the other side

- Used in home environment and on corporate intranets

- Typically, software module on the personal computer

- Easy to configuire

- Used to:
  - deny unauthorized remote access
  - detect and block worms and other malware

Firewall/Bastion Administration

- Access to management console
  - By dedicated clients using encryption
  - Via SSH and https
  - Possibly using also user authentication
- Strategies of disaster recovery
  - Switches capable of Balancing/failover
- Logging
  - Use of a remote syslog server
    - Centralization of all logs
- Security incidents
  - They have different severity levels
  - The policy determines which ones are significant
    - Keep logs for legal analysis about the attacks
    - Synchronization with a time server → important to know which came first

Firewall Topologies

- Host-resident firewall
- Screening router: packet filtering
- Single bastion inline
- Single bastion T, with DMZ
- Double bastion T

Firewall Topologies

- Host-resident firewall
  - personal firewall software and firewall software on servers

- Screening router
  - single router between internal and external networks with stateless or full packet filtering
  - typical for small office/home office (SOHO) applications

Internal network

External network

Screening router
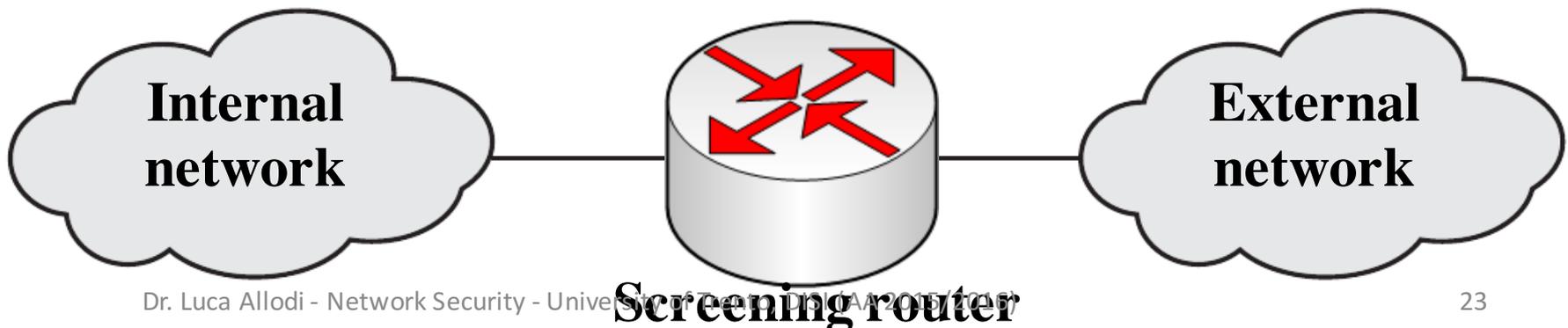
Single Bastion Inline
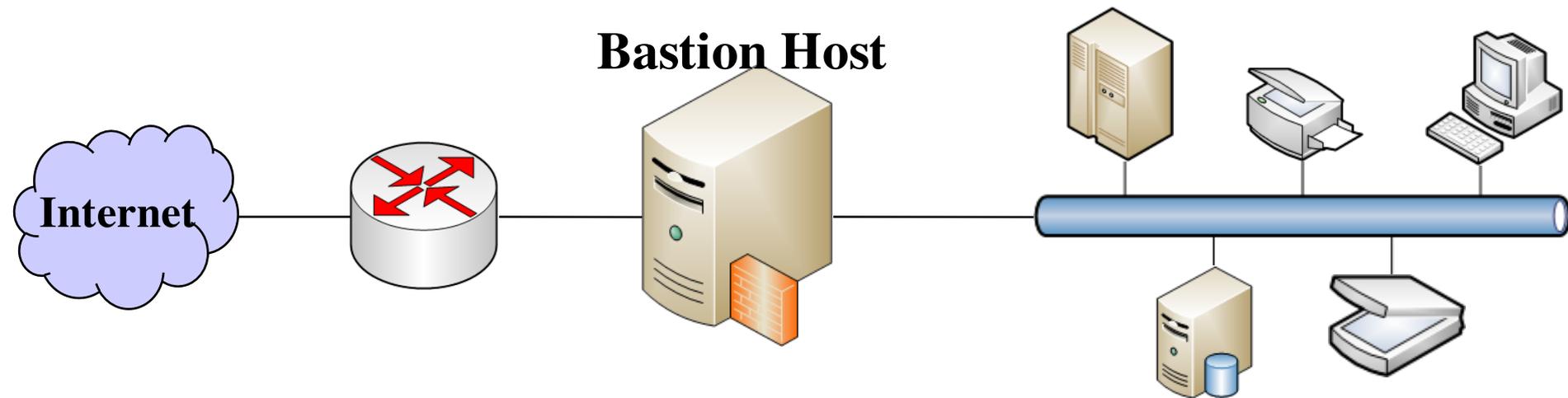
Bastion Host

- Configuration for the packet-filtering router:
  - Only packets from and to the bastion host are allowed to pass through the router
- The bastion host performs authentication and proxy functions
- This configuration implements both packet-level and application-level filtering (allowing for flexibility in defining security policy)
  - An intruder must generally penetrate two separate systems

Single Bastion T

Bastion Host

Internet

DMZ

- Similar to Single Bastion Inline
  - But has third network interface to a DMZ (Demilitarized Zone) where externally visible servers are placed
- Common topology for medium to large organizations
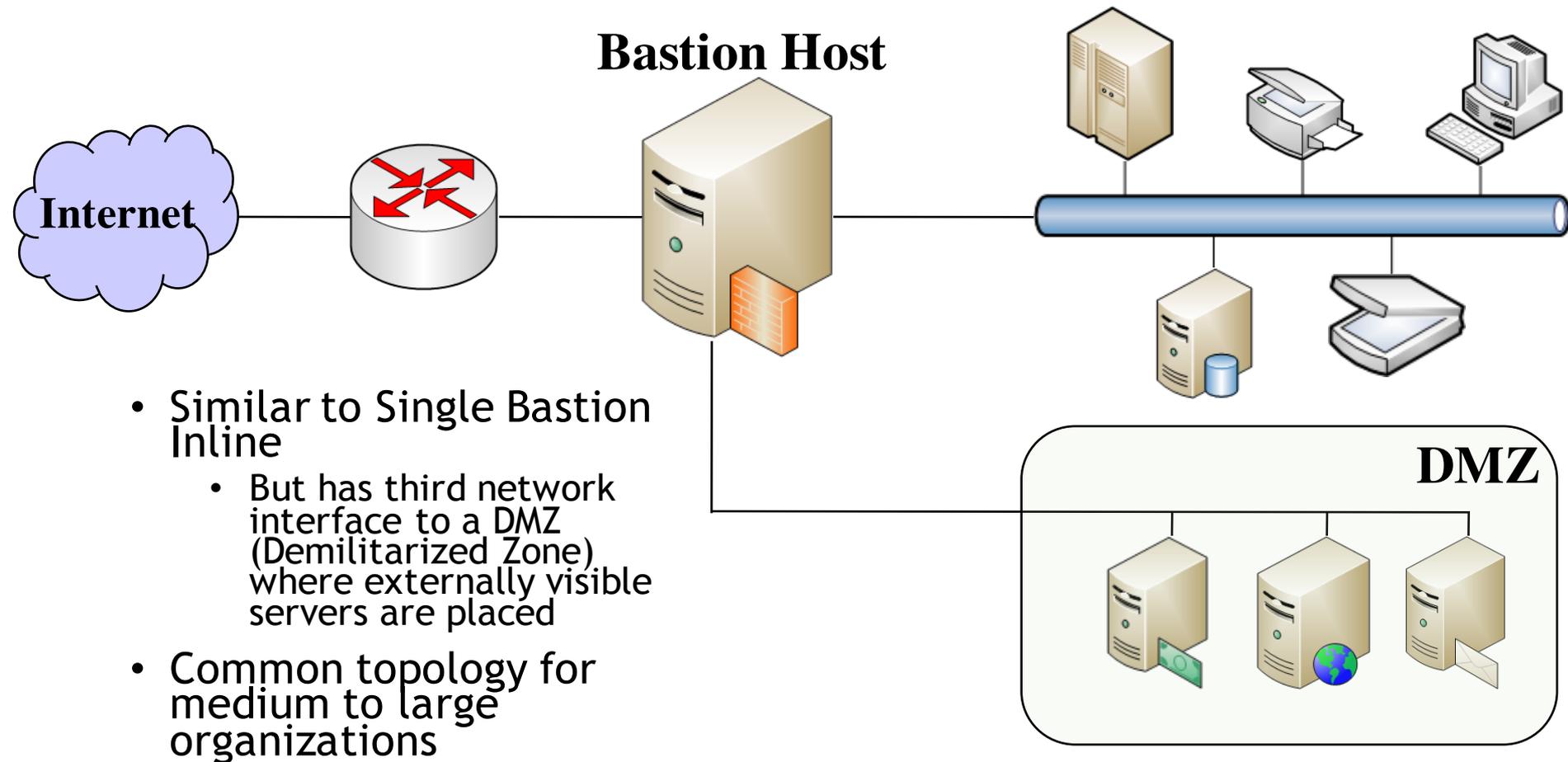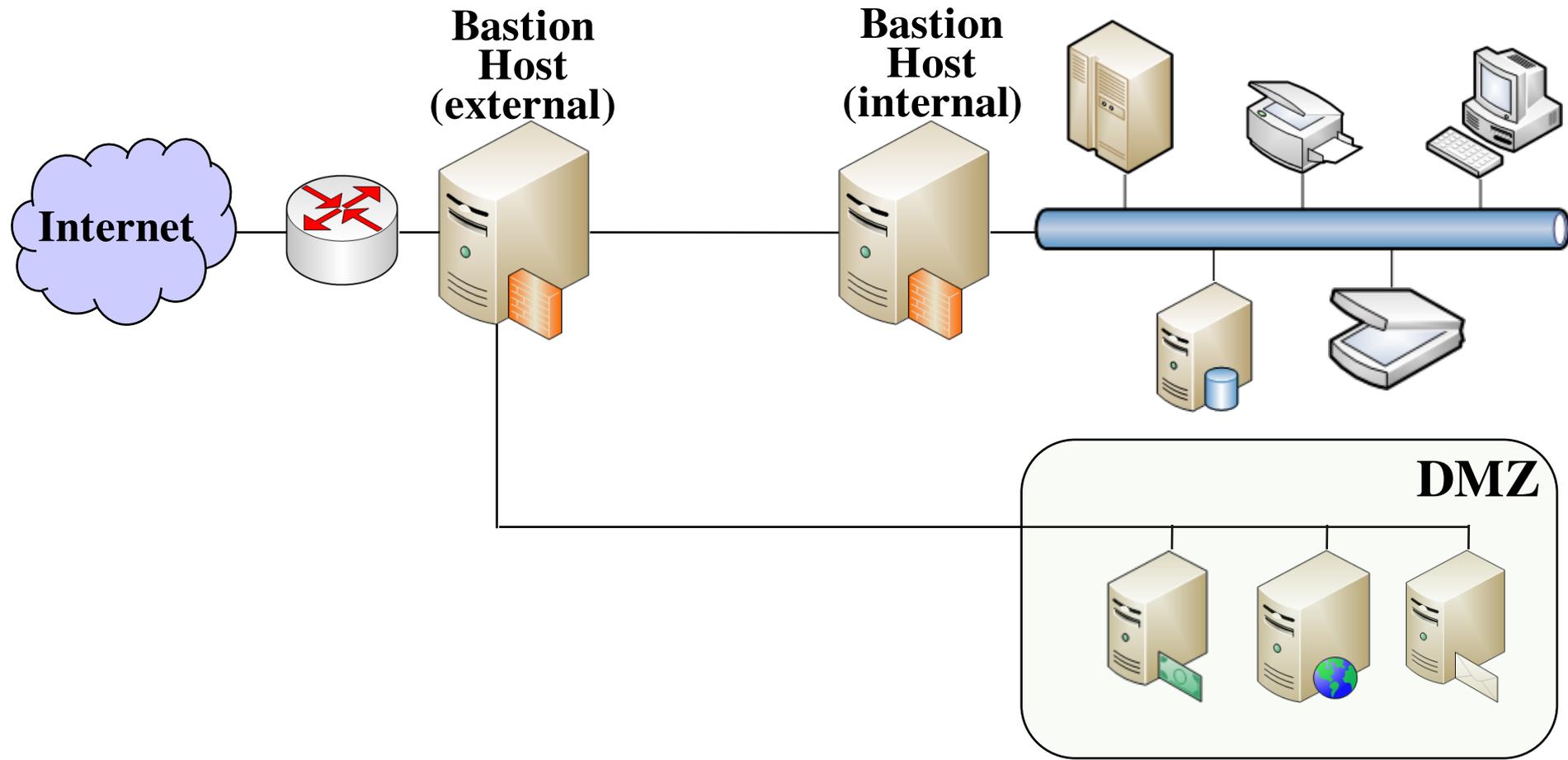
Double Bastion T

Advanced network topologies

- Single/Double bastion topologies are adequate only when mapped to a significant *separation of networks*

- Good network separation allows for
  - Better management of firewall rules
  - Higher control on incoming traffic
  - Higher overall security
  - Lower load on single appliances

Typical multi-level network applications

Internet

Web servers
+
backend

Data servers

Public services
- Front-facing webservers
- Content delivery systems

Application logic
- Backend management
- Dynamic page generation

1st level

Data management
- Databases
- Query backend / data elaboration

2nd level

Separate network topology

Internet

Web servers

Public services
- Front-facing webservers
- Content delivery systems

1st level

Backend

Application logic
- Backend management
- Dynamic page generation

2nd level

Data servers

Data management
- Databases
- Query backend / data elaboration

3rd level

Separate network topology in practice – simple implementation

Internet

Border router
(static filtering)

App Firewall /
Bastion

1st level

3rd level

2nd level

Data servers

Web servers

Backend
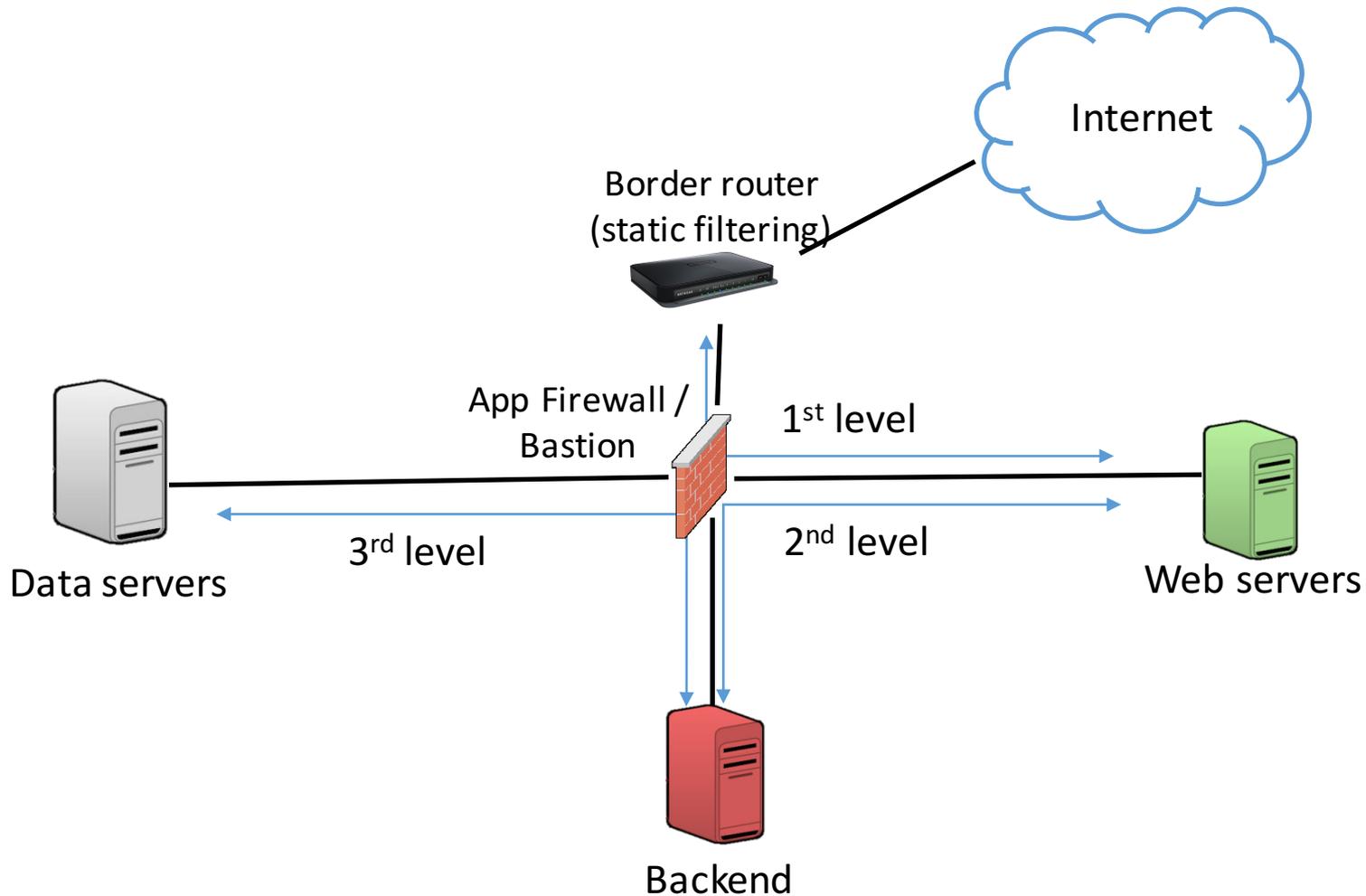
Border router + firewall

- Border router
    - Implements static inward and outward filtering
    - Drop packets toward denied resourcers
        - Best policy → drop with no answer
        - e.g. do not allow packets whose final destination is the firewall
- Firewall
    - Several inward-facing network interfaces
    - Dedicate one interface to each network level
    - Single-point-of-failure
        - Bad configuration may cause network disservices

Divide et impera - Cascade firewalls

Internet

Border router
(static filtering)

Web servers
Non-critical resources

Public services
- Front-facing webservers
- Content delivery systems

Bastion 1st level

- Outward packet filtering
  - e.g. implement client blacklist
- Light application control → Low on resources
  - Prevent DDoS, XSS.

Backend
Critical resources

Application logic
- Backend management
- Dynamic page generation

Bastion 2nd level

- Inward/Outward packet filtering
  - Allow management connections
- Strong application control
  - Deep packet inspection
  - Payload sanitisation

Data servers
Highly critical resources

Data management
- Databases
- Query backend / data elaboration

Bastion 3rd level

- Inward/Outward packet filtering
  - Allow management connections **from trusted network only**
- Strongest application control
 - e.g. Evaluate SQL queries before passing them to mySQL DB

Cascade firewalls - notes

- Inter-dependent firewall policies
- Each firewall must be configured considering functions needed at higher levels
  - E.g. firewall at level 1 must allow all packets eventually directed toward level 2 or 3
  - In complex networks this is unmanageable if network is not well configured
- Requires a good mixture of NAT/PAT policies, firewall configurations, and good separation of services
  - e.g. Hard to have effective NAT + firewalling for SSH services at both level 1 and level 3 → where should the packet go?
    - Remember incoming packet will always have address of outward-facing NAT interface toward port 22.
  - Each layer should ideally be in a different subnet
    - Firewall @ Layer 1: 192.168.1.0/24
    - Firewall @ Layer 2: 192.168.2.0/24, etc..
    - ✓ F1 Accept all traffic that needs to be forwarded to F2
- High design, management, maintenance costs
  - Introducing a new service at any level requires testing all configuration at lower levels

Divide et impera - Parallel firewalls

Internet

Border router
(static filtering)

Web servers

Public services
- Front-facing webservers
- Content delivery systems

Application logic
- Backend management
- Dynamic page generation

Backend

Data management
- Databases
- Query backend / data elaboration

Data servers
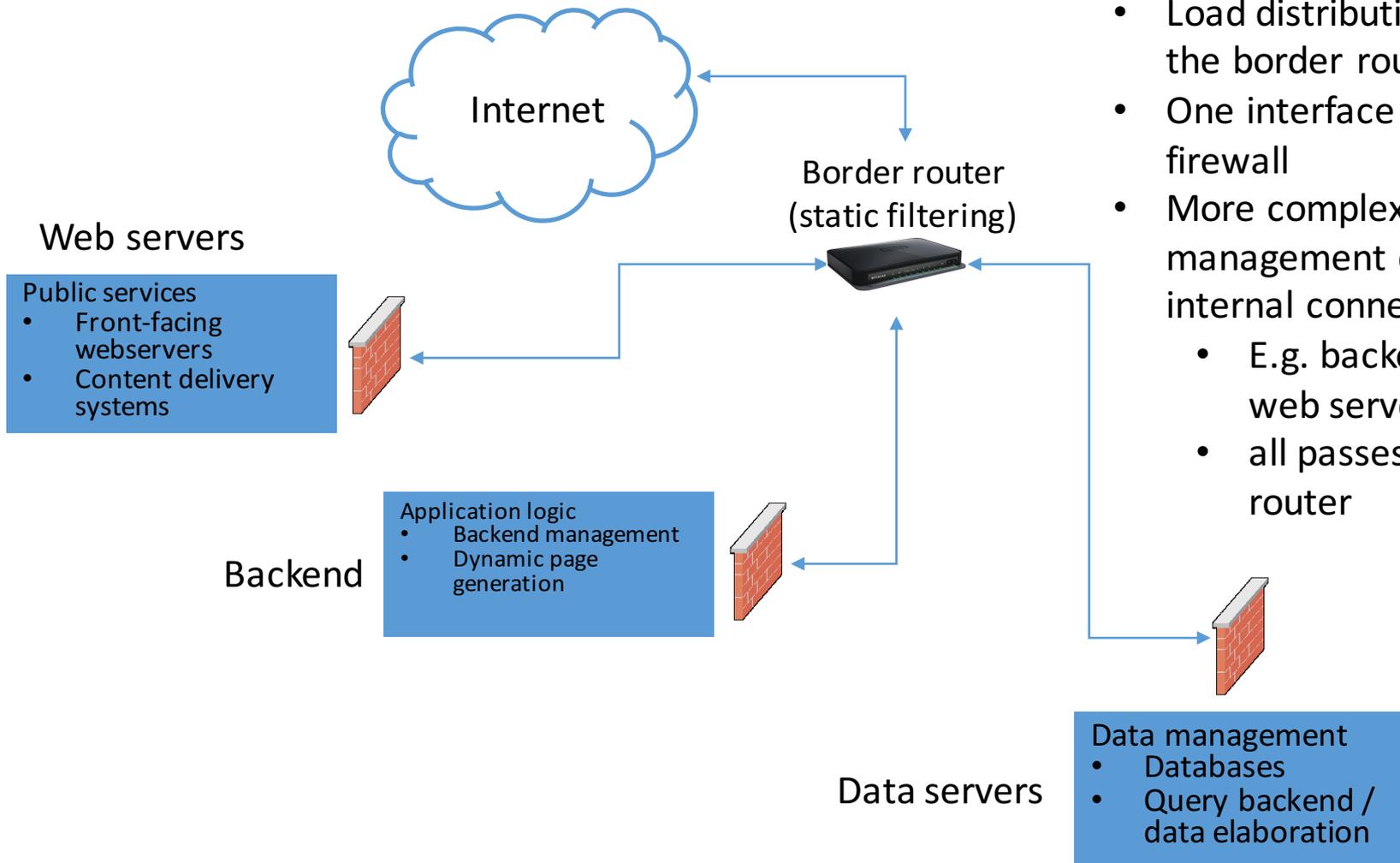
- Load distribution is on the border router
- One interface per firewall
- More complex management of internal connections
  - E.g. backend → web servers
  - all passes through router

Mixed architectures

Internet

Border router
(static filtering)

Web servers

Public services
- Front-facing webservers
- Content delivery systems

172.16.0.0/16

- More flexible management
- Adding systems to intranet requires only configuration of second level router
  - only responsible firewall's policies are affected
- Strong network separation still possible

Second level Router

172.17.0.0/16

Intranet

Application logic
- Backend management
- Dynamic page generation

Backend   172.17.1.0/24

Internal clients

172.17.2.0/24

Data management
- Databases
- Query backend / data elaboration

Data servers