UNIVERSITÀ DEGLI STUDI
DI TRENTO

Network Security

AA 2015/2016

System hardening

(Application Firewalls, IDS)

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
```

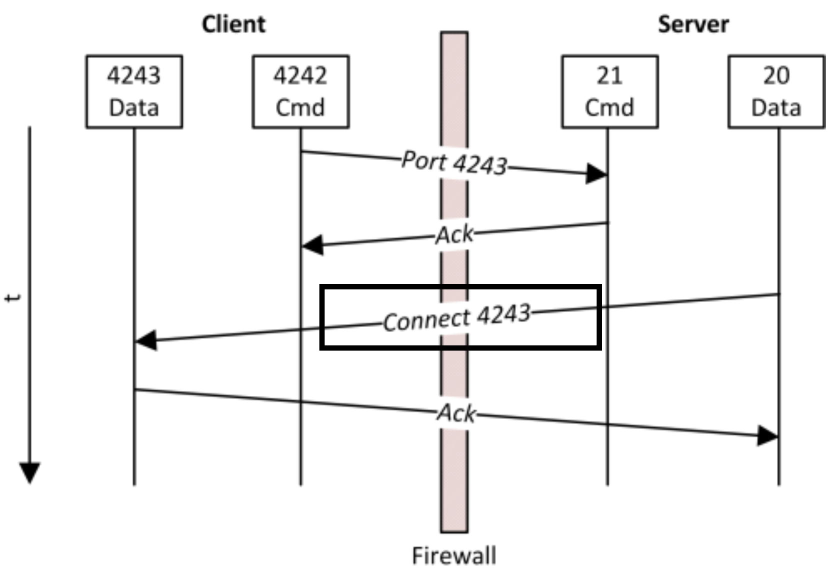
- Mixed rules also possible:

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

UNIVERSITÀ DEGLI STUDI DI TRENTO

Application firewalls

- Staefulness consider also application layer
  - “Deep packet inspection”
  - Can keep track of and deny others
  - e.g. FTP PORT command



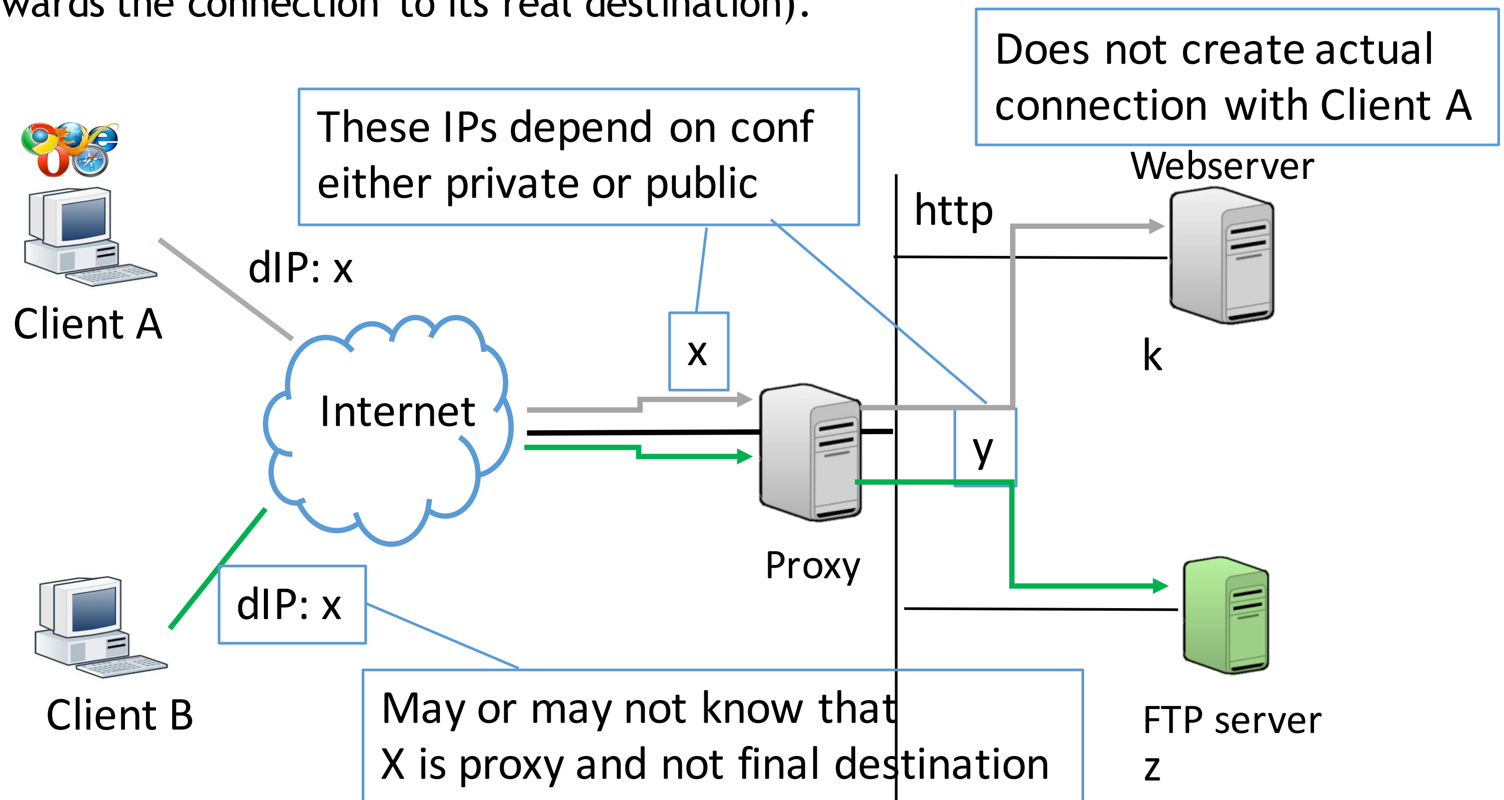
- FTP commands are passed to port 21
- In “Active mode” the server opens a connection with the client, and chooses dport
  - this happens with PORT command
- Application firewall can detect PORT command and act on packet
  - Simple stateful firewall can not easily manage this

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”

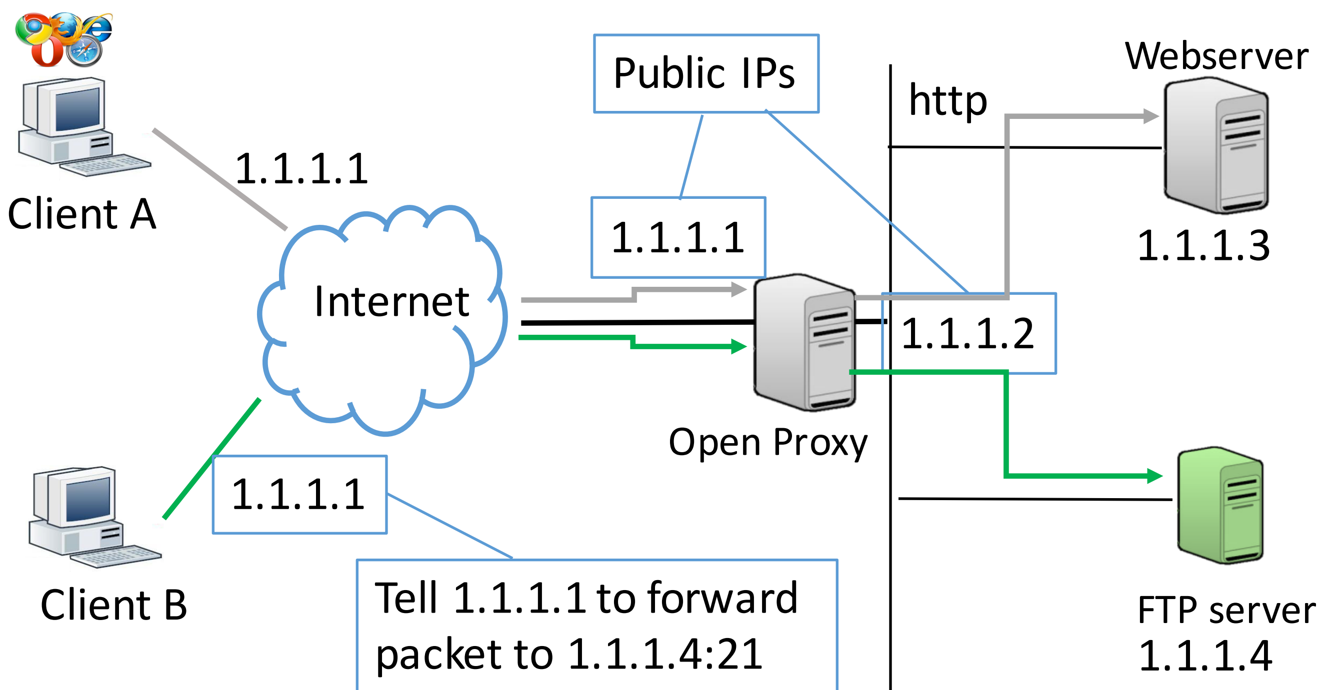
Proxy

- 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).



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

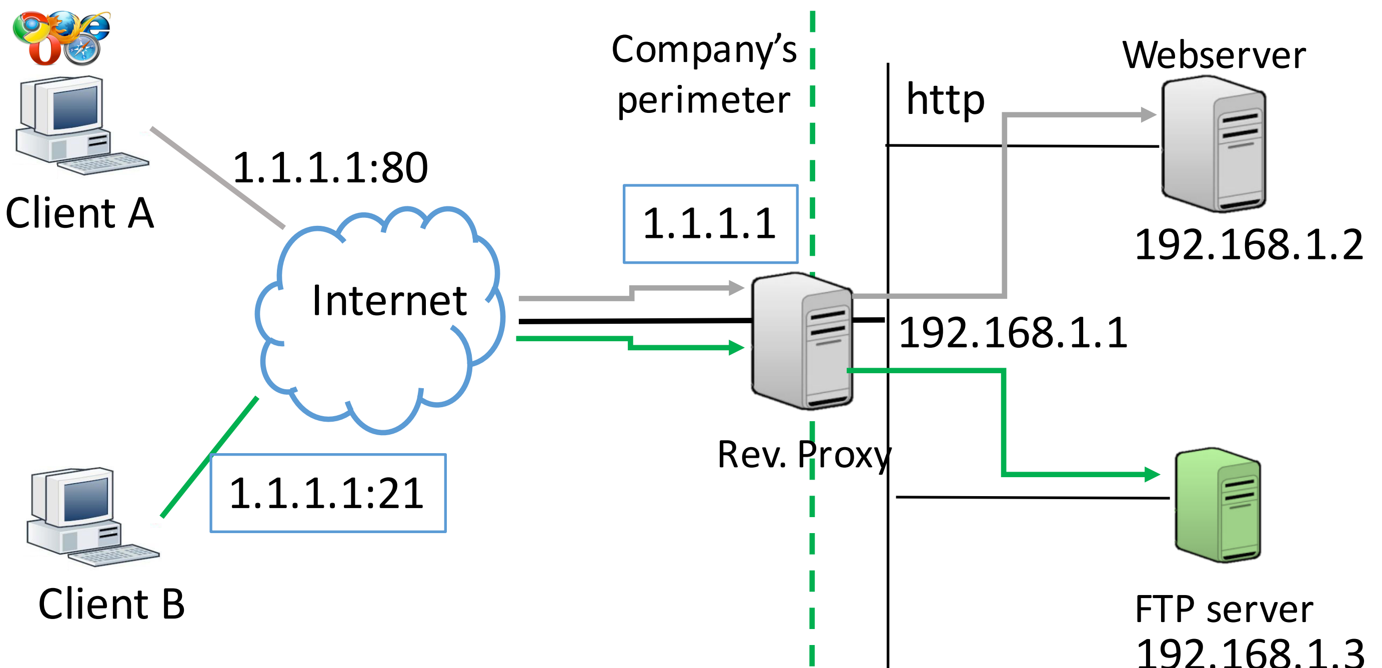


Open proxy - characteristics

- 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.

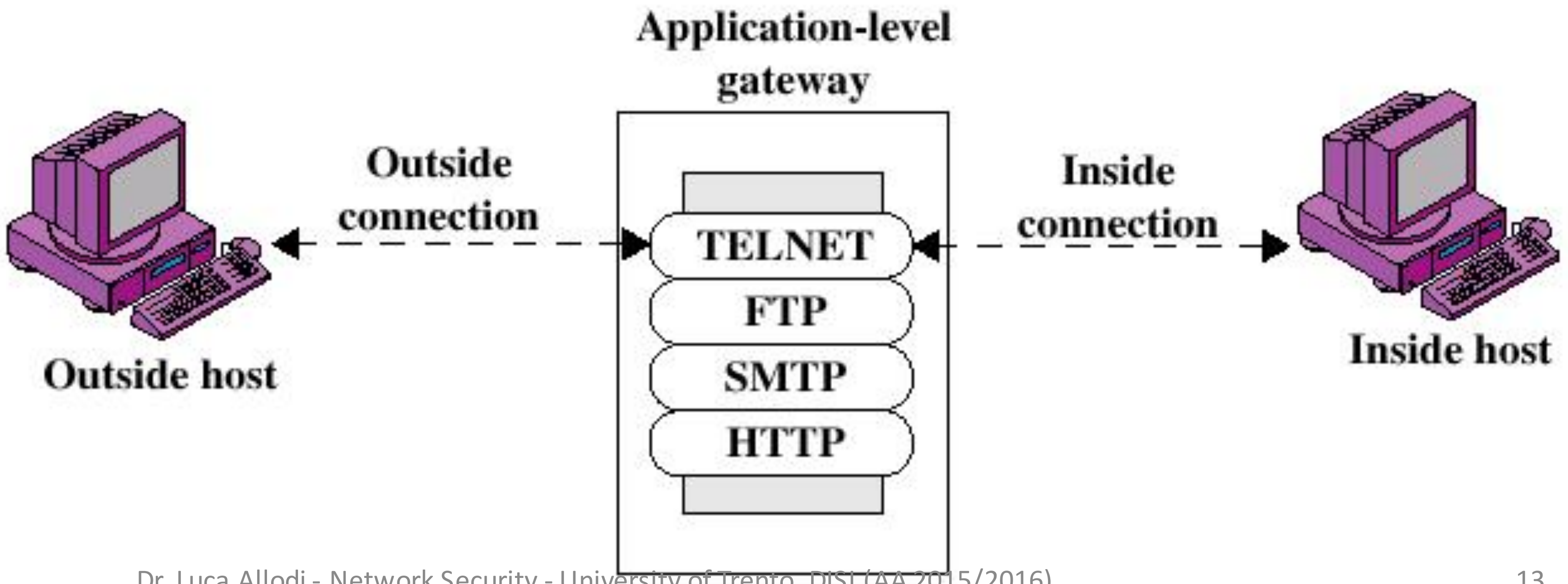


Reverse proxy - characteristics

- 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

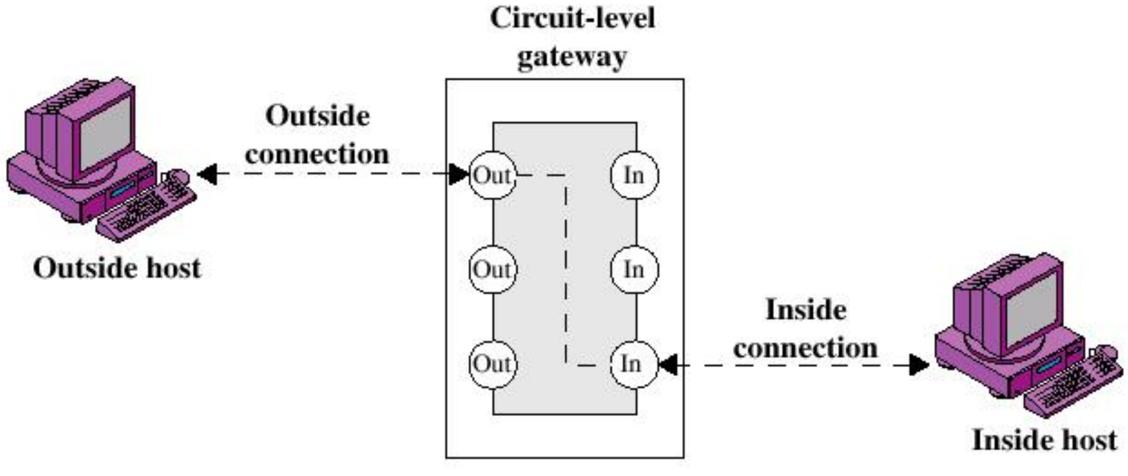


Application Gateway: Pros

- Advantages: by not 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



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

Firewall/Bastion Administration

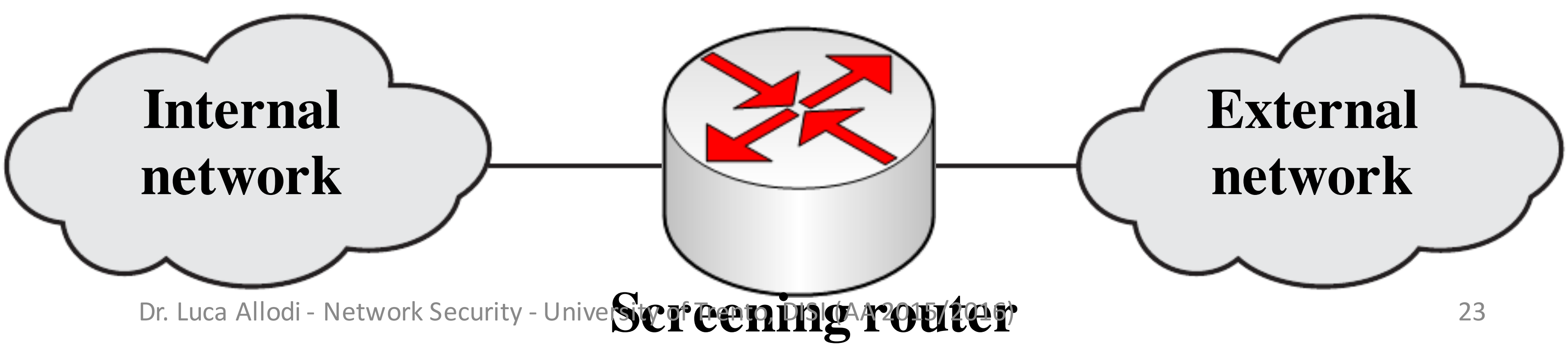
- Access to management console
  - By dedicated clients using encryption
  - Via SSH and https
  - Possibly using also user authentication
- Strategies of disaster recovery
  - Switches capable of Balancing/failover
- Logging
  - Use of a remote syslog server
    - Centralization of all logs
- Security 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

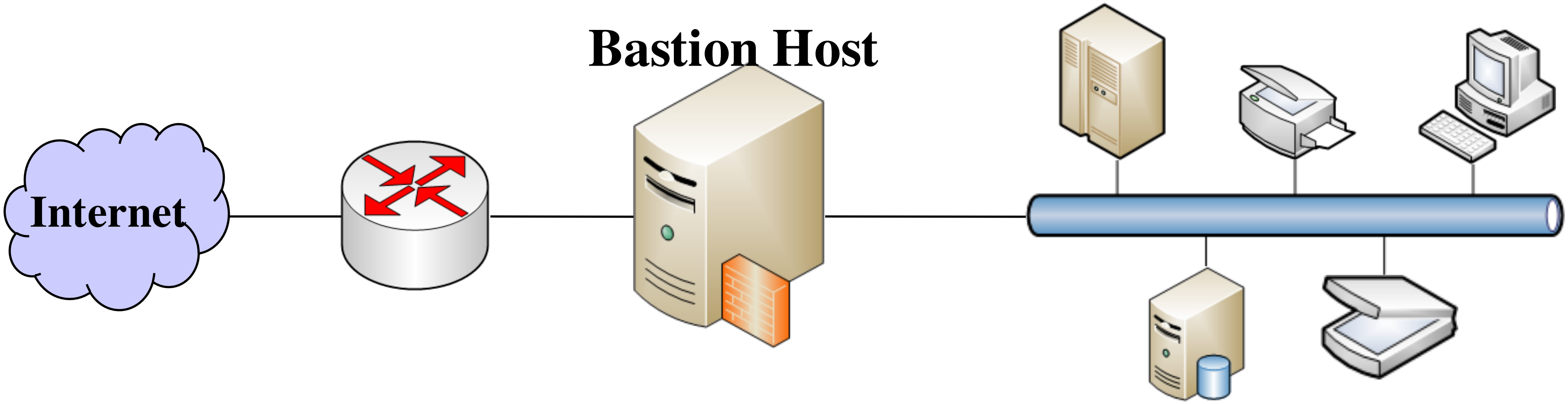
- Host-resident firewall
- Screening router: packet filtering
- Single bastion inline
- Single bastion T, with DMZ
- Double bastion T

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

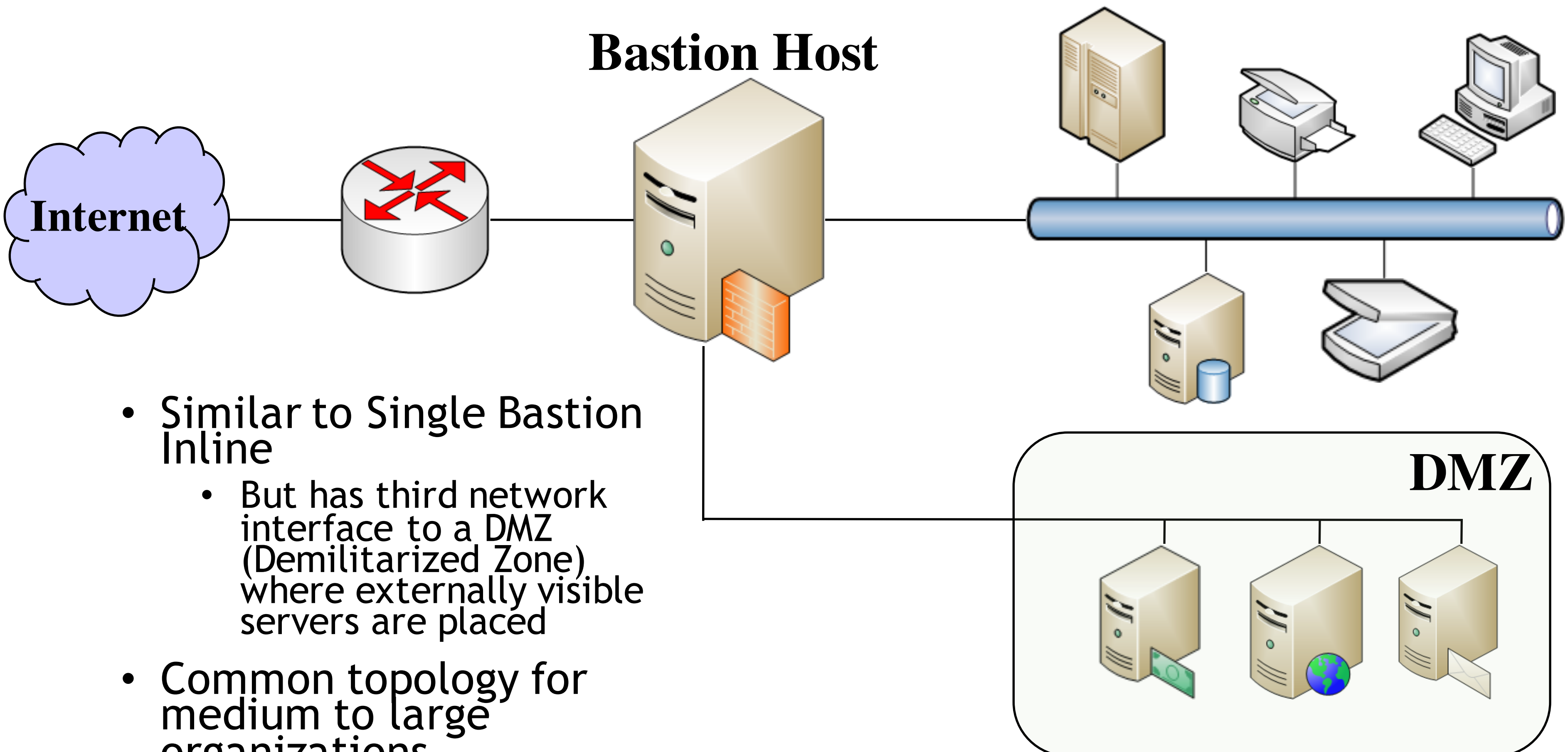


Single Bastion Inline



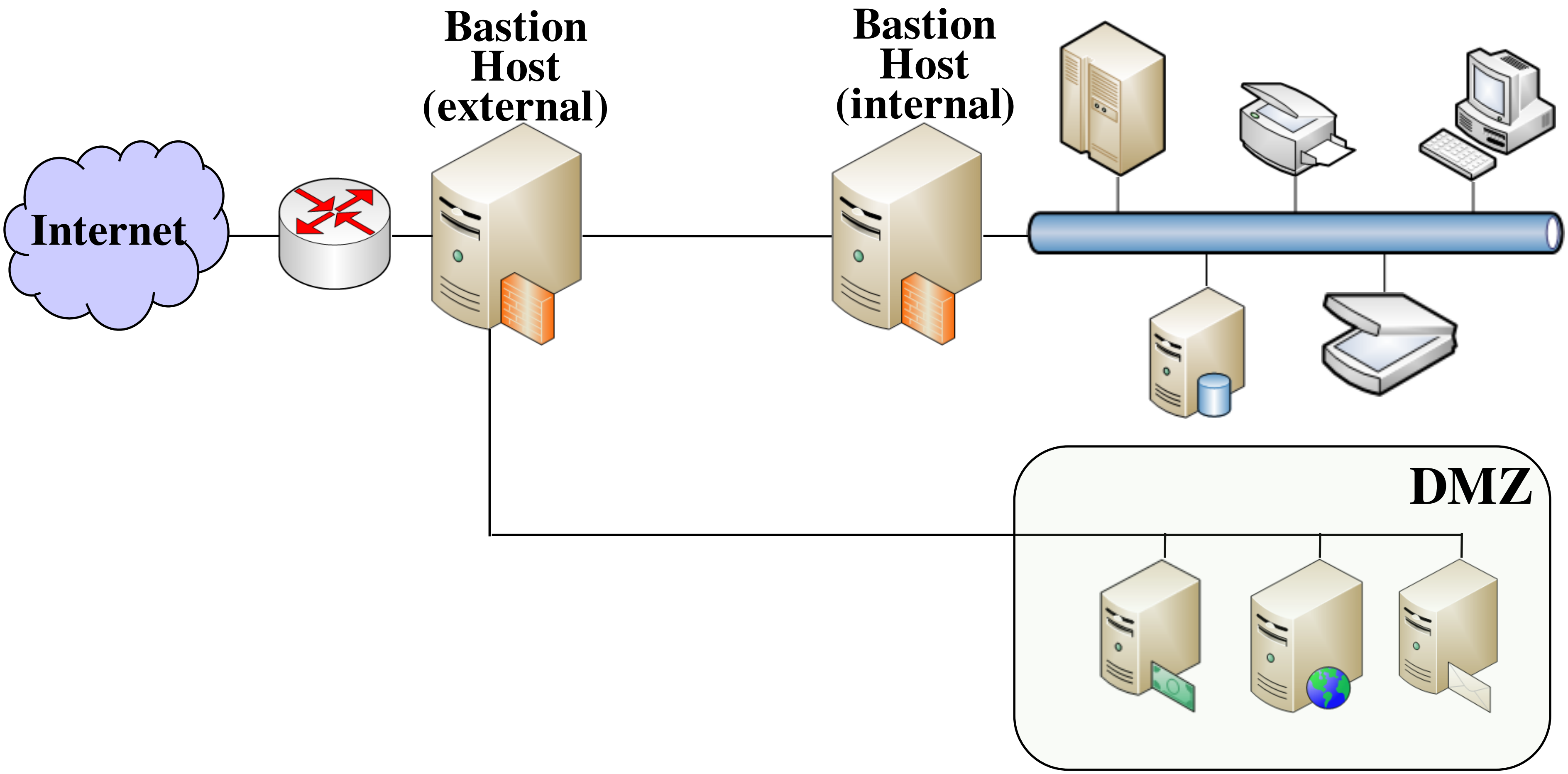
- Configuration for the packet-filtering router:
  - Only packets from and to the bastion host are allowed to pass through the router
- The bastion host performs authentication and proxy functions
- This configuration implements both packet-level and application-level filtering (allowing for flexibility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

Single Bastion T



- 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

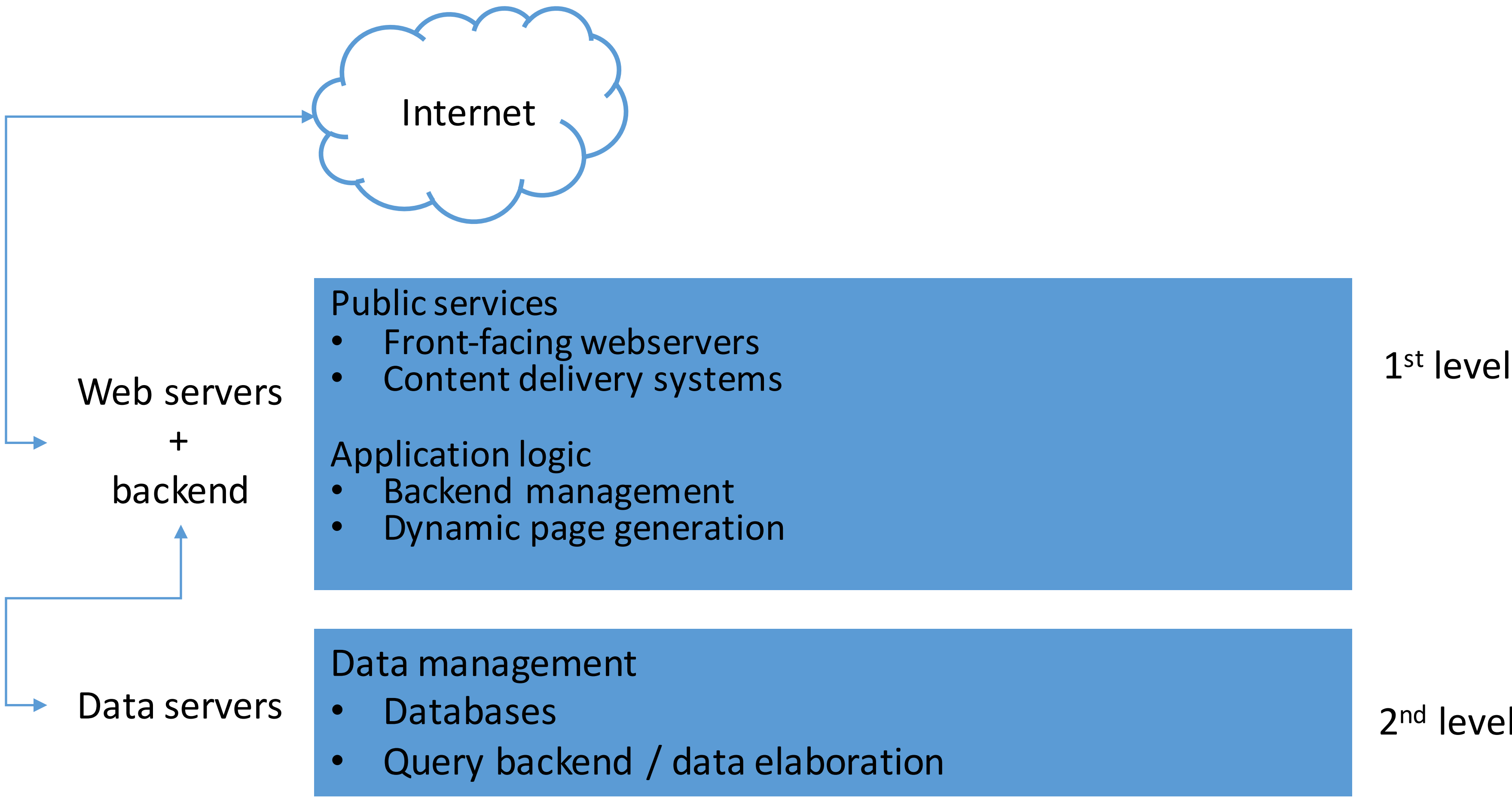
Double Bastion T



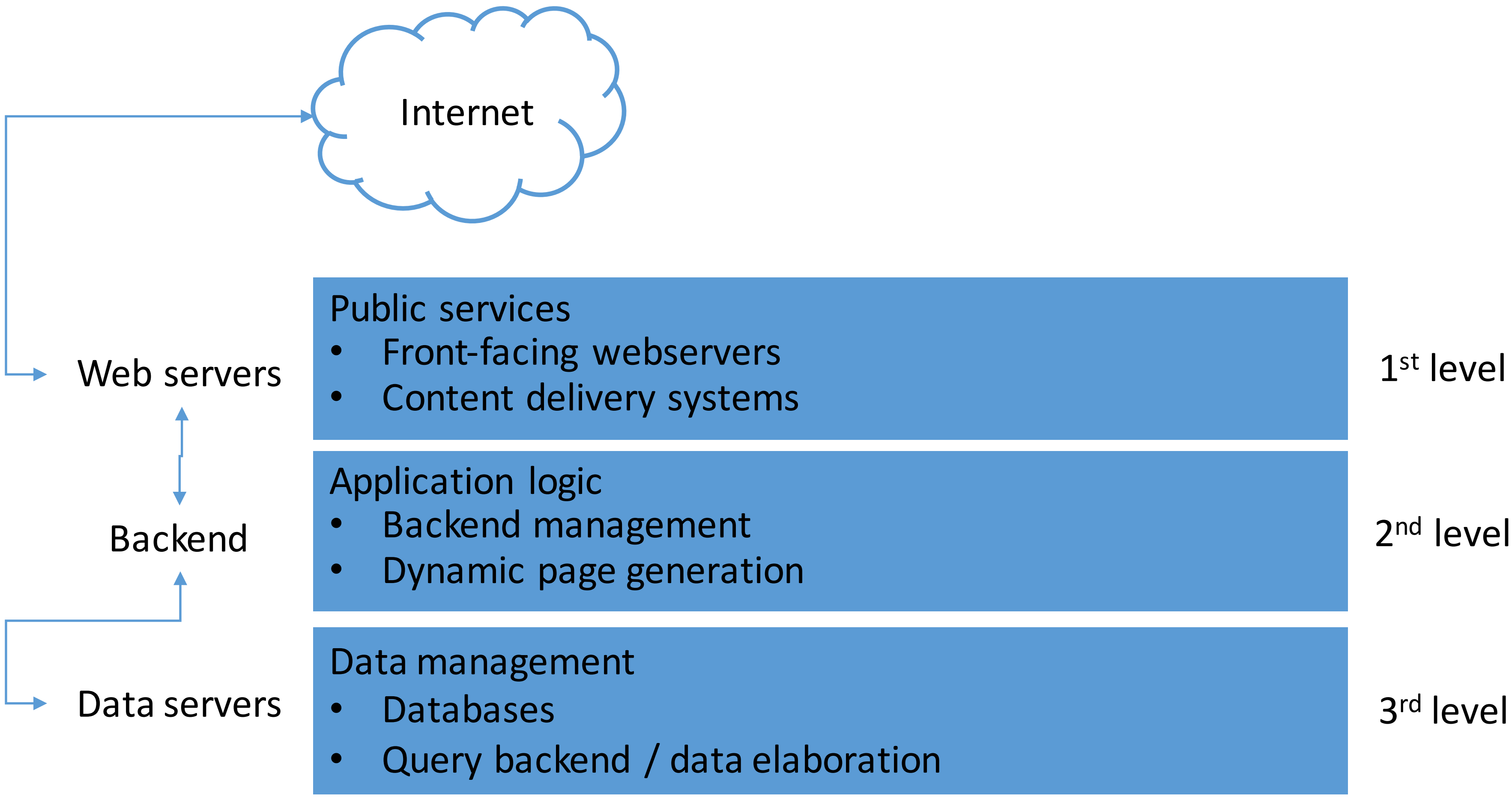
Advanced network topologies

- 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

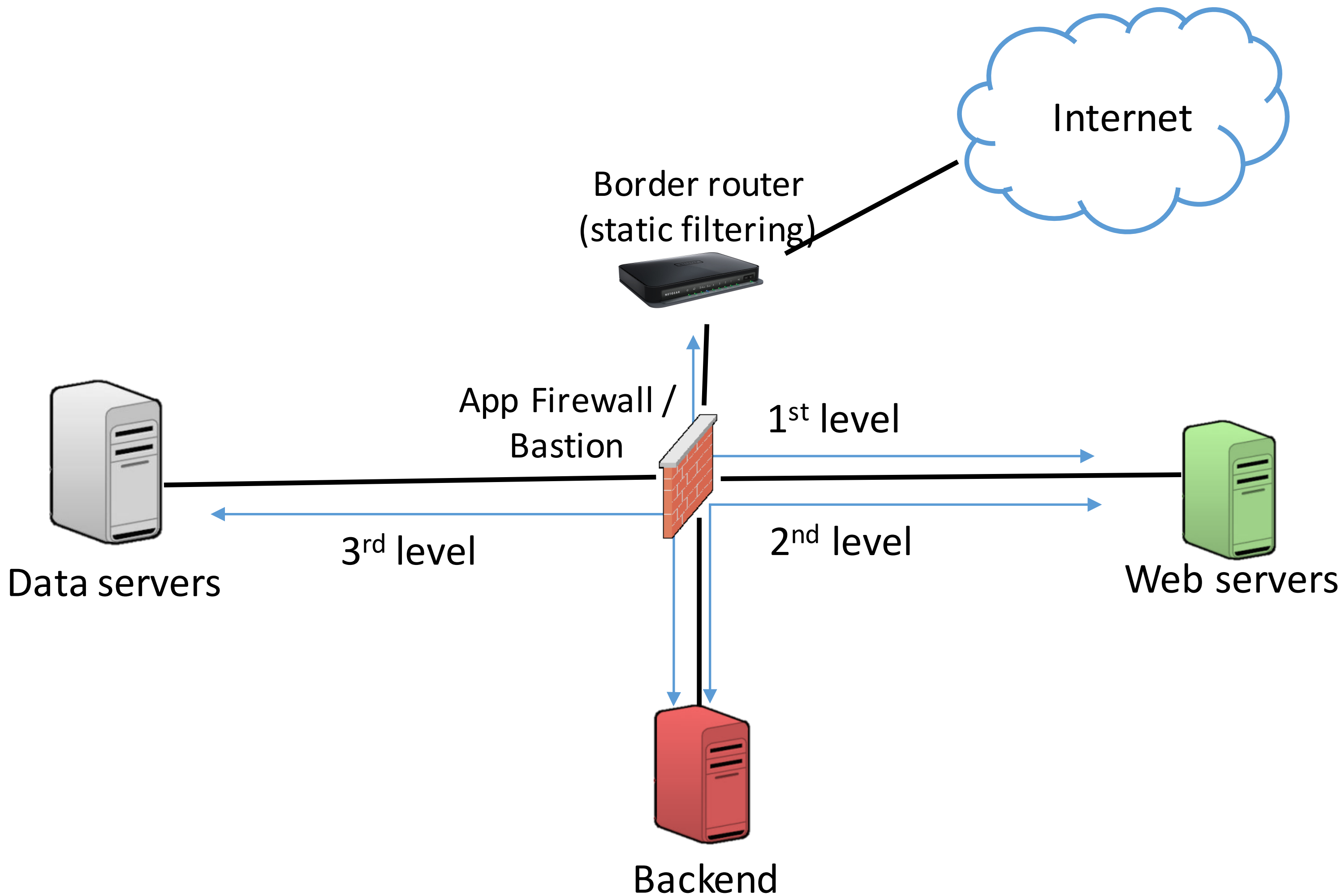
Typical multi-level network applications



Separate network topology



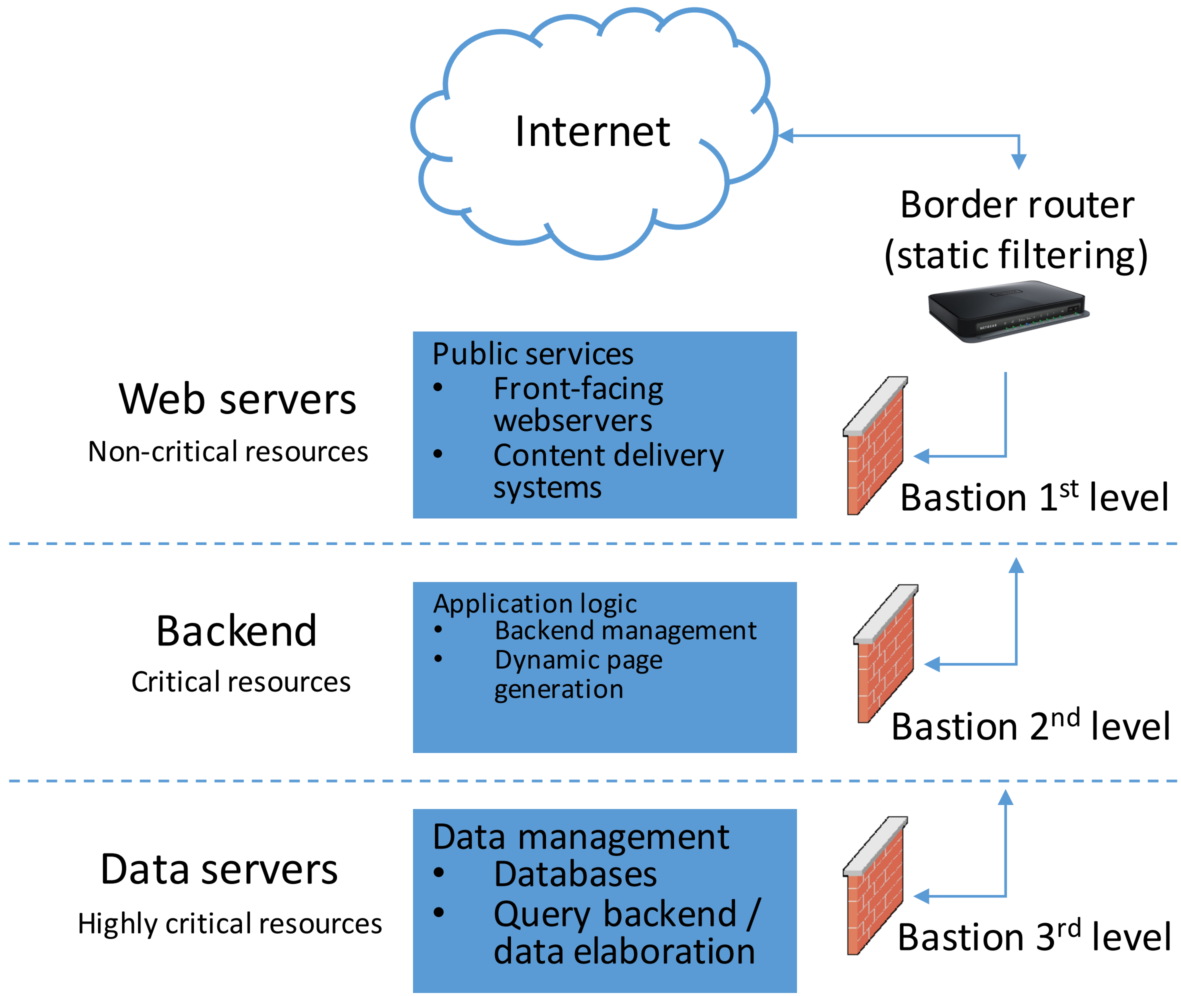
Separate network topology in practice – simple implementation



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



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

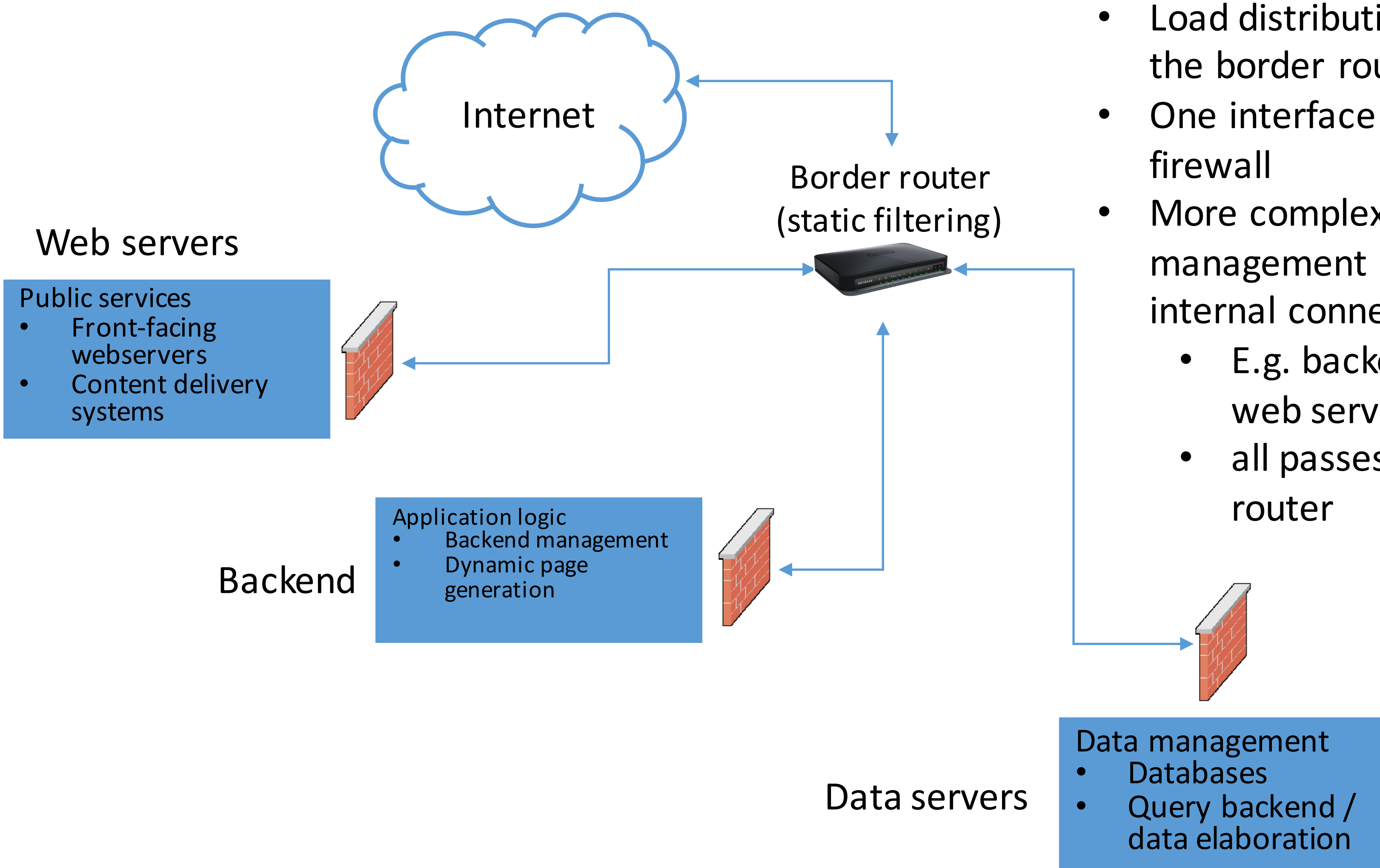
- Inward/Outward packet filtering
  - Allow management connections
- Strong application control
  - Deep packet inspection
  - Payload sanitisation

- 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

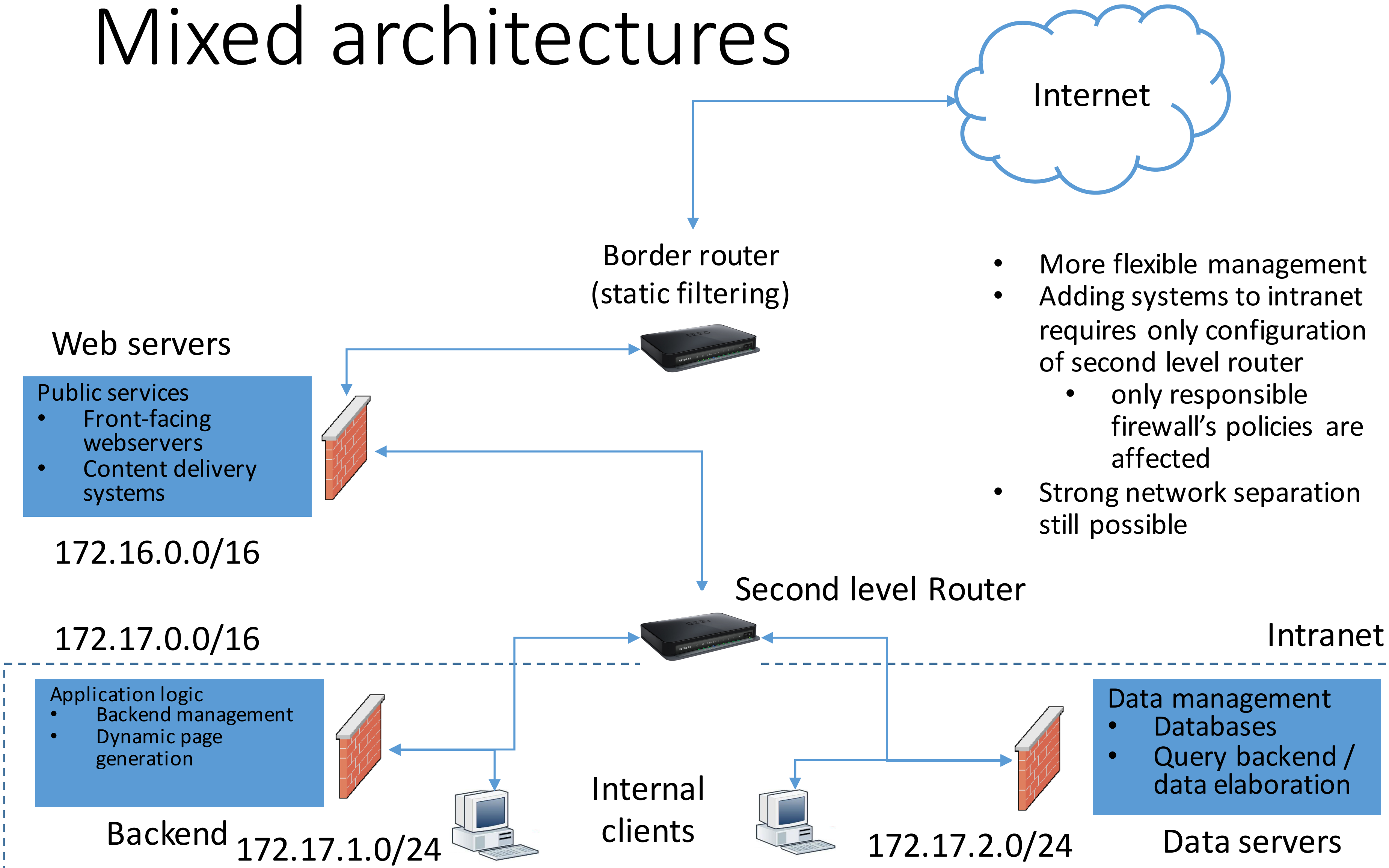
- Inter-dependent firewall policies
- Each firewall must be configured considering functions needed at higher levels
    - E.g. firewall at level 1 must allow all packets eventually directed toward level 2 or 3
    - In complex networks this is unmanageable if network is not well configured
- Requires 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



- 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



- 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

Intrusion Detection Systems

Function of an IDS

- Firewalls prevent unwanted access to network resources that should be isolated w.r.t. another network
- IDS monitors incoming connections
  - Depending on its position in the network may provide different functionalities
    - More on this later
- Intrusion Prevention Systems (IPS) can act over "malicious" behaviour
- IDS → passive monitoring
- IPS → active monitoring
- In reality functionalities are not entirely distinct
  - Commercial lingo rather than actually different technology

IDS – 3 phases

1. Data collection
   - Host-based IDS → Sit on an host (client, server)
   - Network-based IDS → Collects network data
2. Data analysis
   - Two distinct approaches
   - Misuse detection →list unwanted behaviour, report if detected
   - Anomaly detection → build average profile, report if current activity significantly different from average
3. Action
   - IDS → report, log entry
   - IPS → report, log entry, block/alert

Misuse detection

- IDS equivalent of "default allow" policies
- "blacklist" patterns that are believed to be related to malicious activities
  - System calls
  - Payloads in network protocols
- Signature-based
  - Very diffused detection technique
  - Easy to deploy
  - Typical implementation for network-based IDSs
- As all blacklisting approaches (signature-based) it can only detect patterns that are *already known*

Anomaly detection

- Assumes intruder behaviour differs from legitimate profile
- Building legitimate profile may be an issue
  - Depends on data used for profiling (e.g. sampled vs whole dataset)
  - Profile can evolve → new "legitimate activity" looks suspicious
- Can be used both for HIDS and NIDS
  - HIDS → syscall, system file hashing, system states, ..
  - NIDS → protocol analysis, similar to application proxy
    - Monitoring as opposed to filtering

Network IDS

- Baseline implementation is of type *misuse detection*
  - Easier to implement
  - Network traffic is hard to predict even on well-controlled environments
- Signature example:

```
alert
tcp $EXTERNAL_NET any -> $HOME_NET 139
flow:to_server,established
content:"|eb2f 5feb 4a5e 89fb 893e 89f2|"
msg:"EXPLOIT x86 linux samba overflow"
reference:bugtraq,1816 reference:cve,CVE-1999-0811
```

The base-rate fallacy – or, can we have actually good detection rates?

- Both anomaly and misuses detection necessarily lead to false positives and false negatives
- A NIDS with 99% true positive rate and 99% true negative rate seems to have high-reliability alarms
  - → an alarm fires up → you should worry
  - → no alarm fires up → all is good
  - But is it?
- Base-rate fallacy
  - Simple derivation from Bayes theorem
  - Very well known by medics and doctors
  - Still making its way through in InfoSec

The base-rate fallacy [Axelsson 2000]

- Tests with high true positives and negatives rates yield much “worse” results than expected by the average user
- Remember Bayes theorem

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{\sum_{i=1}^{n} P(A_i) \cdot P(B|A_i)}$$

This is P(B) expanded to all “n” cases for A that B comprises

- Let’s make the classic medical example
  - Attack = illness
  - IDS Alarm = medical test

Base-rate fallacy example

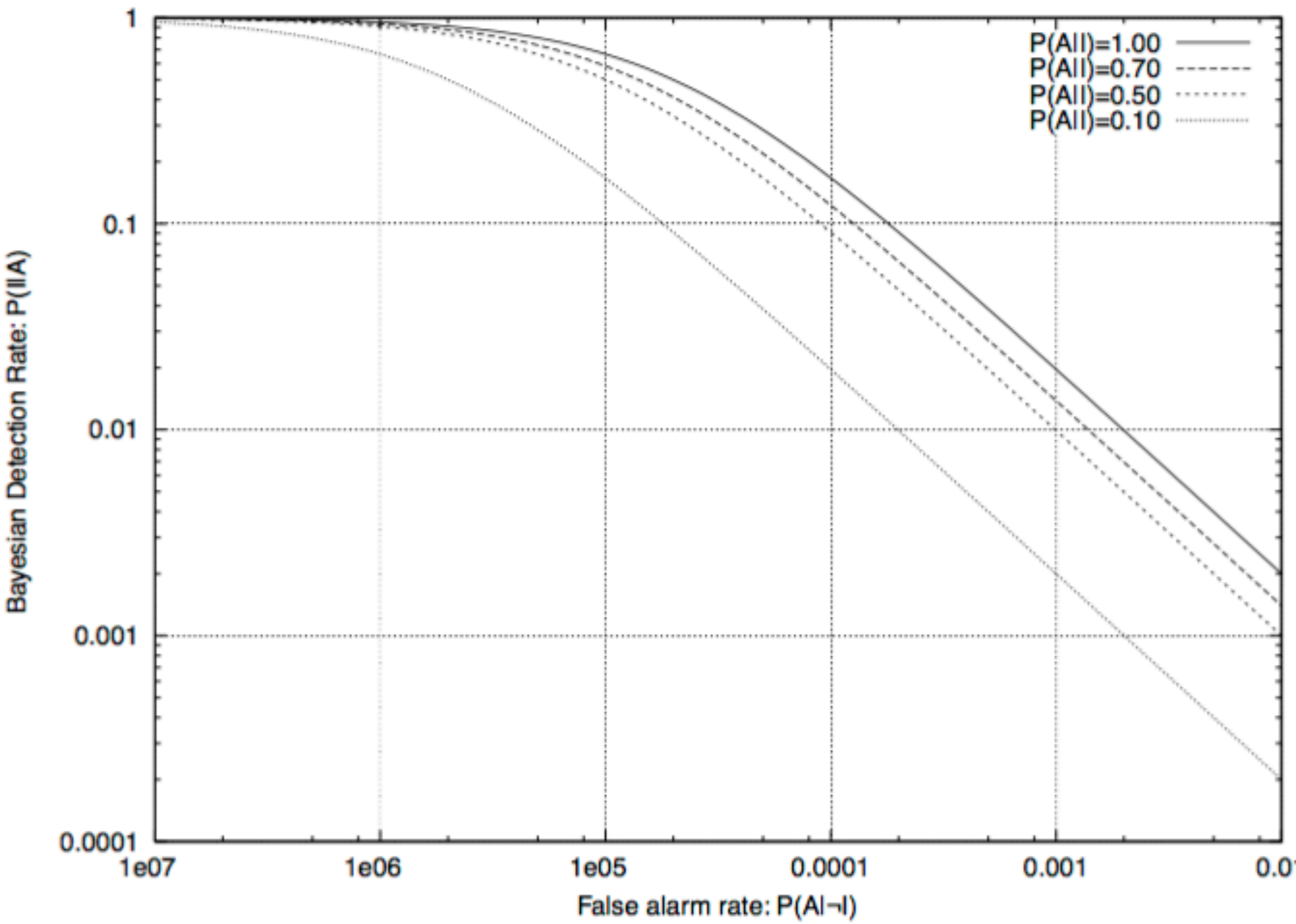
$$P(A|B) = \frac{P(A) \cdot P(B|A)}{\sum_{i=1}^{n} P(A_i) \cdot P(B|A_i)}$$

- A=event is *patient is sick*
- B=medical test says patient is sick
- P(A|B) = patient is actually sick given that test said so
  - Equivalent to "there is an actual attack given that NIDS fired alarm"
- Set TP=99%; TN=99% → P(B|A) = 0.99
- Diseases are rare. Say 1/10.000 people have the illness → P(A)=1/10.000
  - Most network traffic is legitimate

$$P(A|B) = \frac{1/10000 \cdot 0.99}{1/10000 \cdot 0.99 + (1 - 1/10000) \cdot 0.01} = 0.00980\ldots \approx 1\%$$

- There is only 1% chance that patient is sick when test says so
  - An alarm is not very meaningful → IDS alarms are hard to manage → log analysis

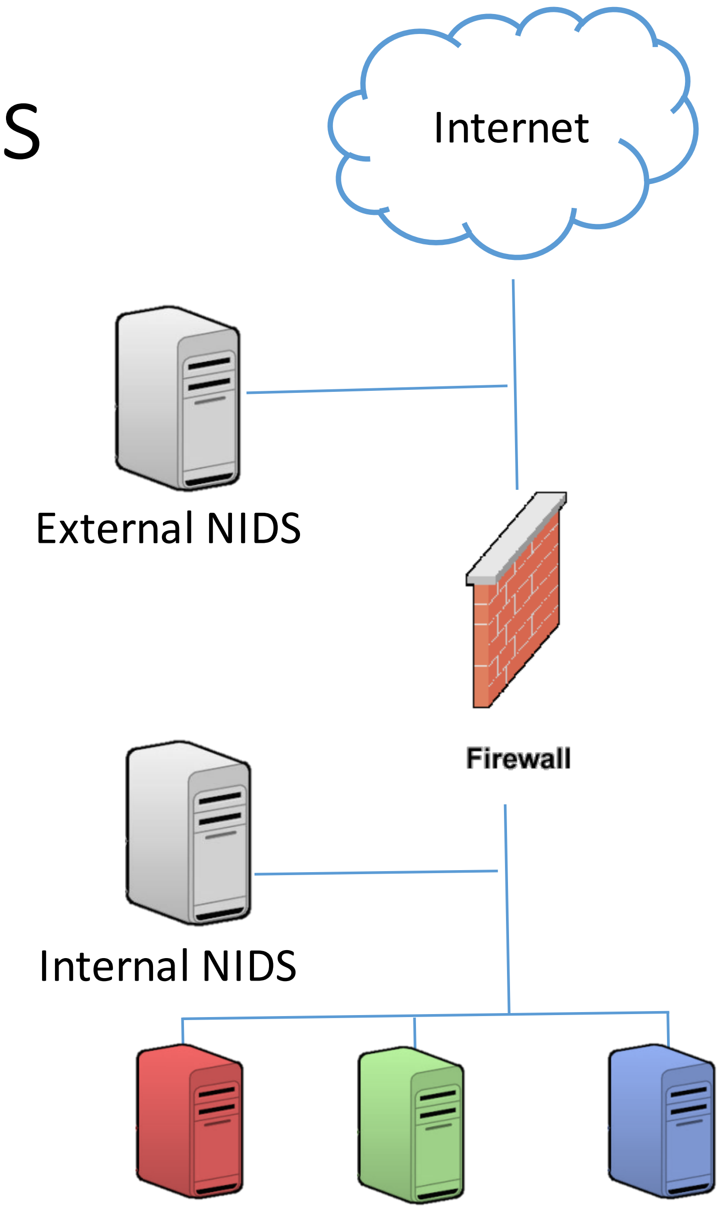
Base-rate fallacy and IDSs



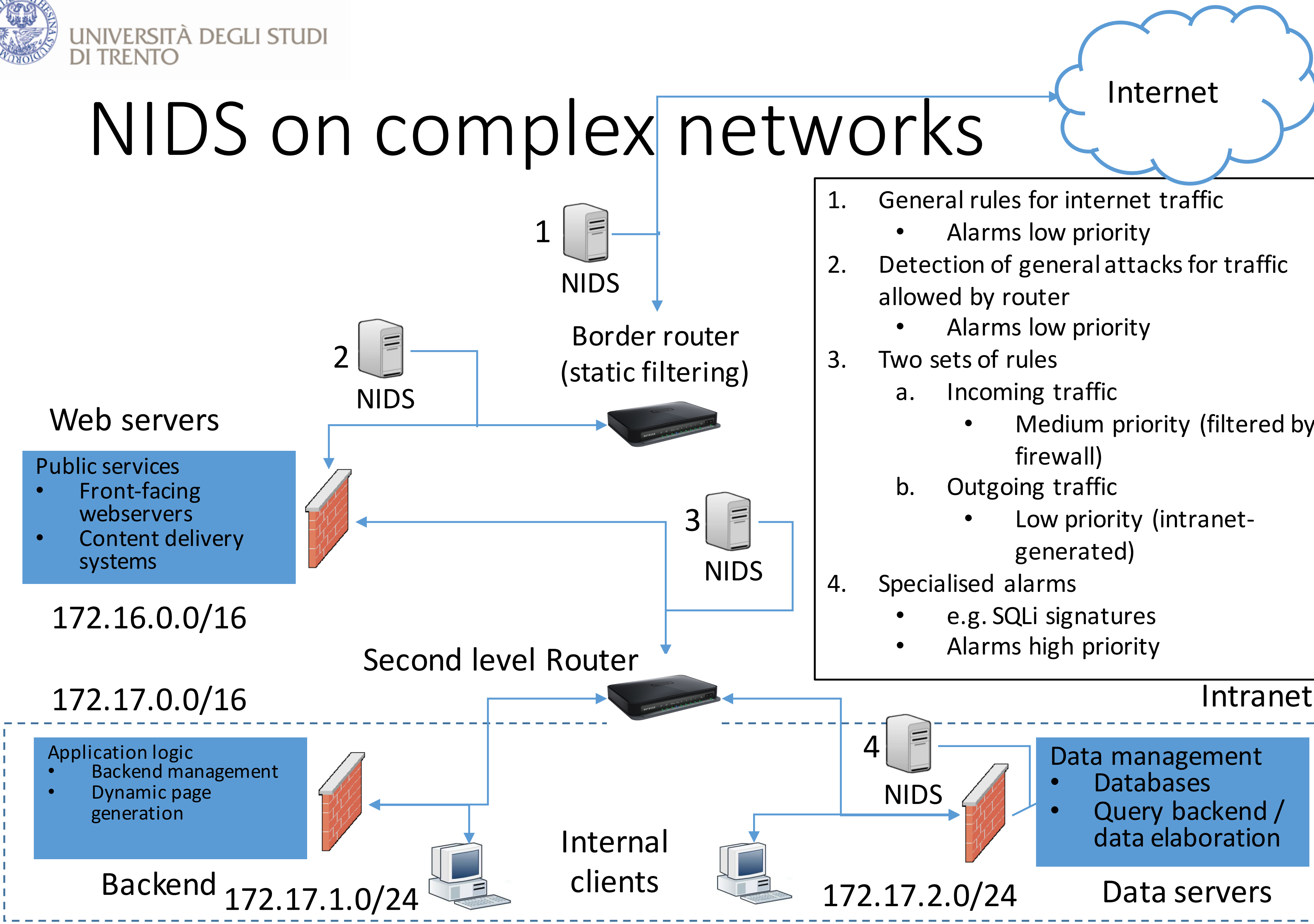
Notice that the false positives rate is the one that dominates the curve

Architectural aspects

- External NIDS
  - Analysis of all set of incoming traffic
  - Only general signatures are possible
    - high incidence of FP
  - All detected “attempted attacks” are logged
- Internal NIDS
  - Analysis of traffic allowed by the firewall
  - More specific signatures are possible
    - e.g. based on services behind firewall, subnet characteristics, ..
  - Says nothing about attacks attempted but blocked by firewall



NIDS on complex networks



1. General rules for internet traffic
   - Alarms low priority
2. Detection of general attacks for traffic allowed by router
   - Alarms low priority
3. Two sets of rules
   a. Incoming traffic
      - Medium priority (filtered by firewall)
   b. Outgoing traffic
      - Low priority (intranet-generated)
4. Specialised alarms
   - e.g. SQLi signatures
   - Alarms high priority

NIDS evasion [Siddharth 2005]

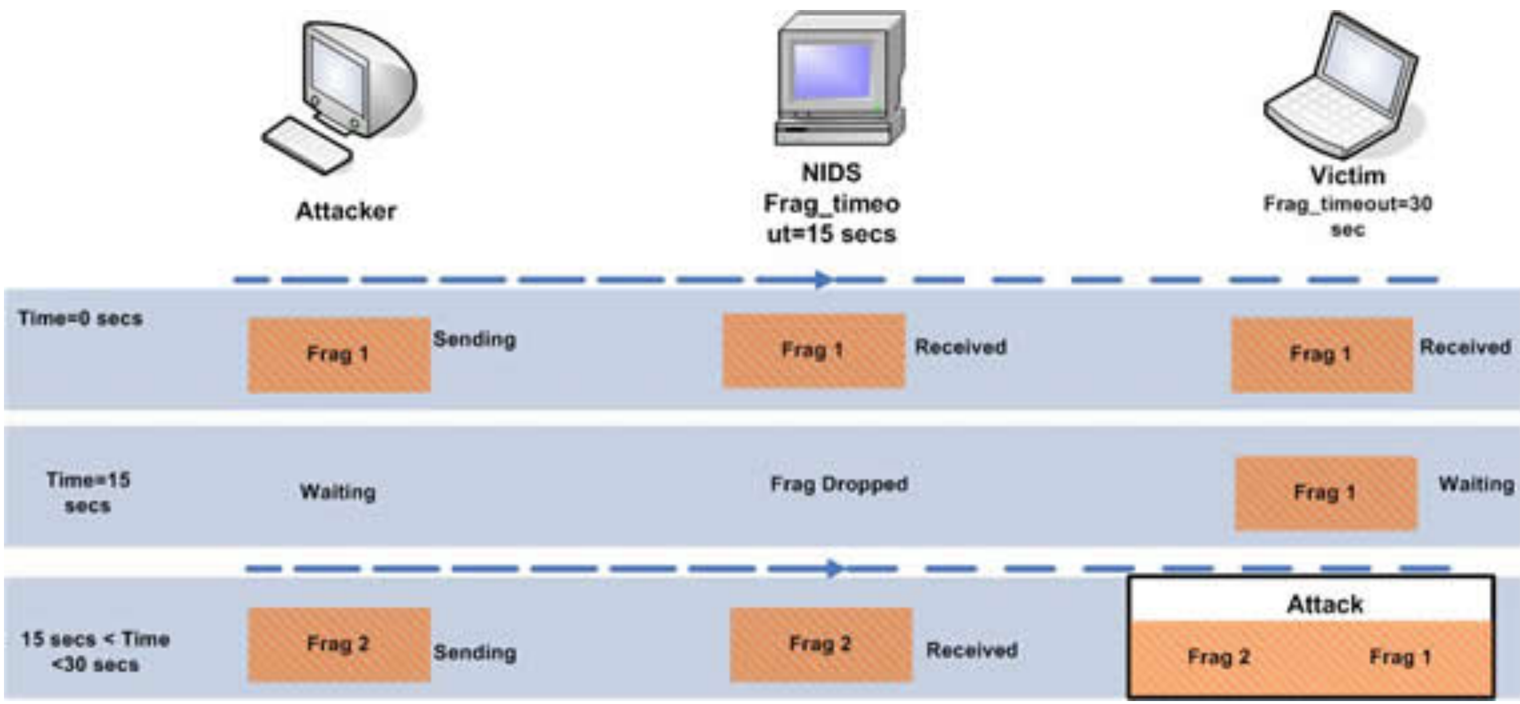
- Signature-based evasion can be fairly trivial
- Depends on implementation of actual signature

`content:”/bin/bash”`

- → detects remote calls to bash
- Does not detect string “/etc/../bin/bash”, etc.
- More advanced techniques are typically based on IP fragmentation
  - All techniques have common goal: NIDS sees different packet than client
  - Look at these keeping in mind you may want to prevent the attacker from performing
    - Network mapping
    - OS fingerprinting

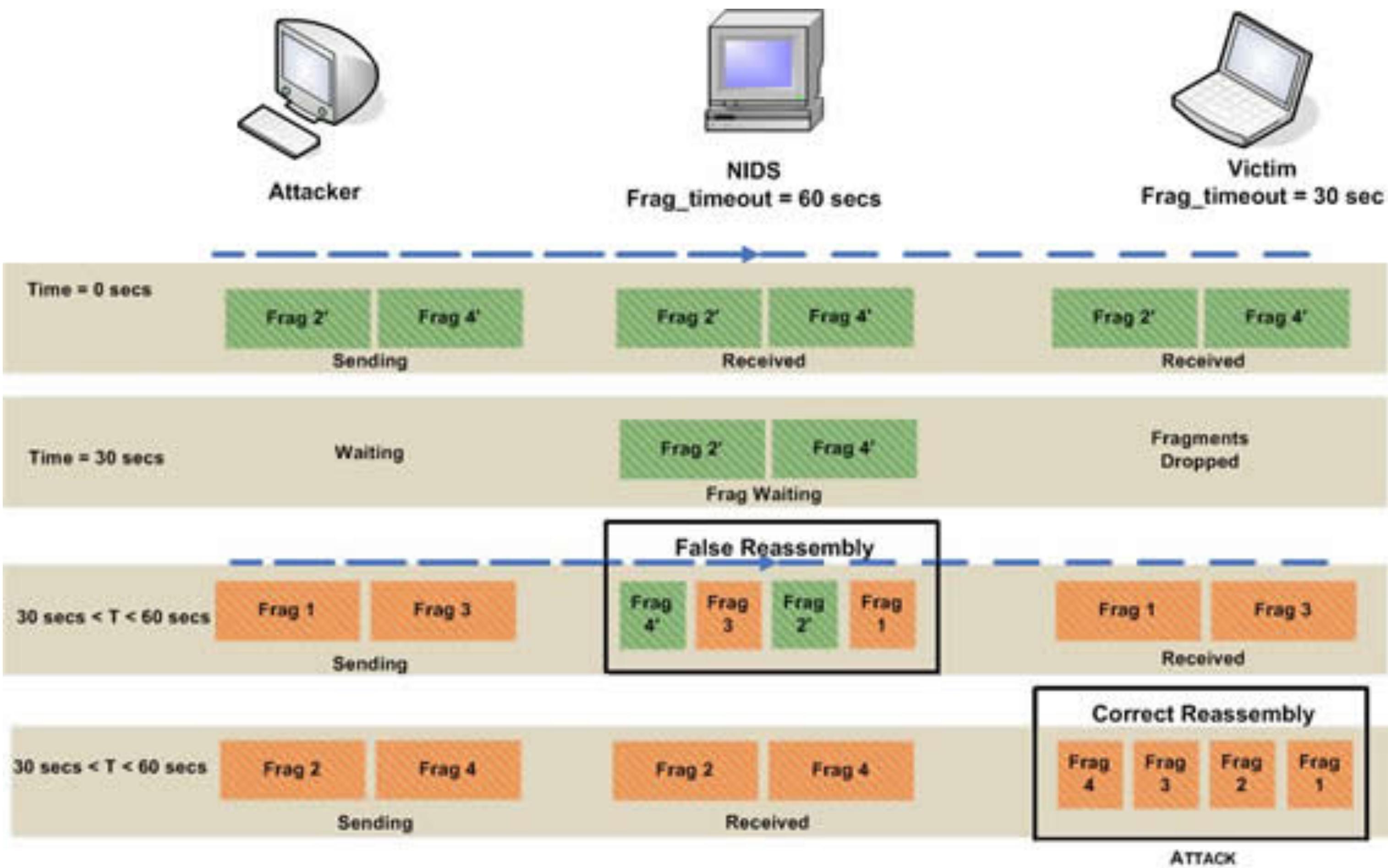
Evasion technique – Riassembly time-out

- NIDS has lower riassembly timeout than receiving client



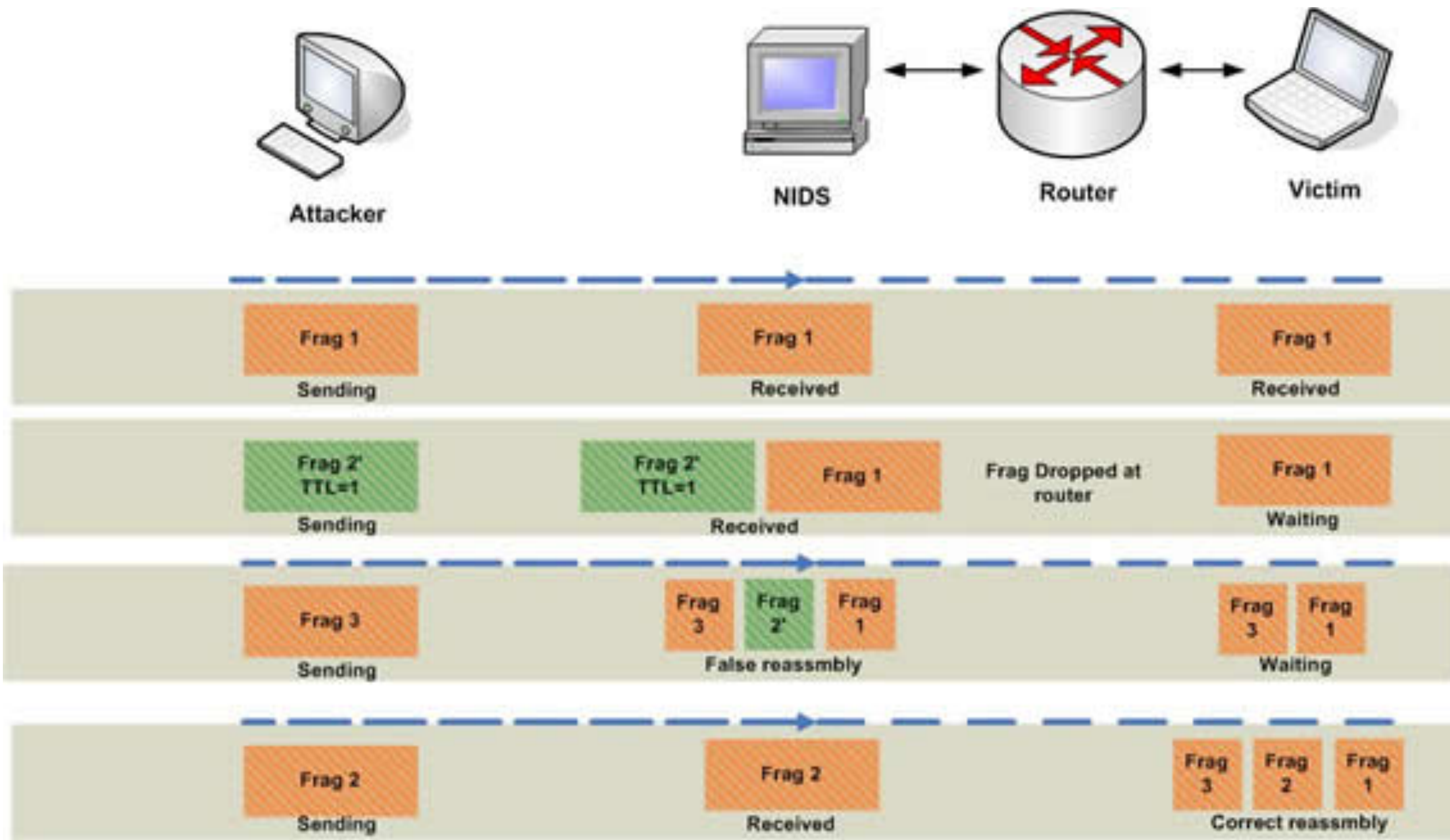
Evasion technique – Riassembly time-out (2)

- NIDS has <u>higher</u> riassembly timeout than receiving client



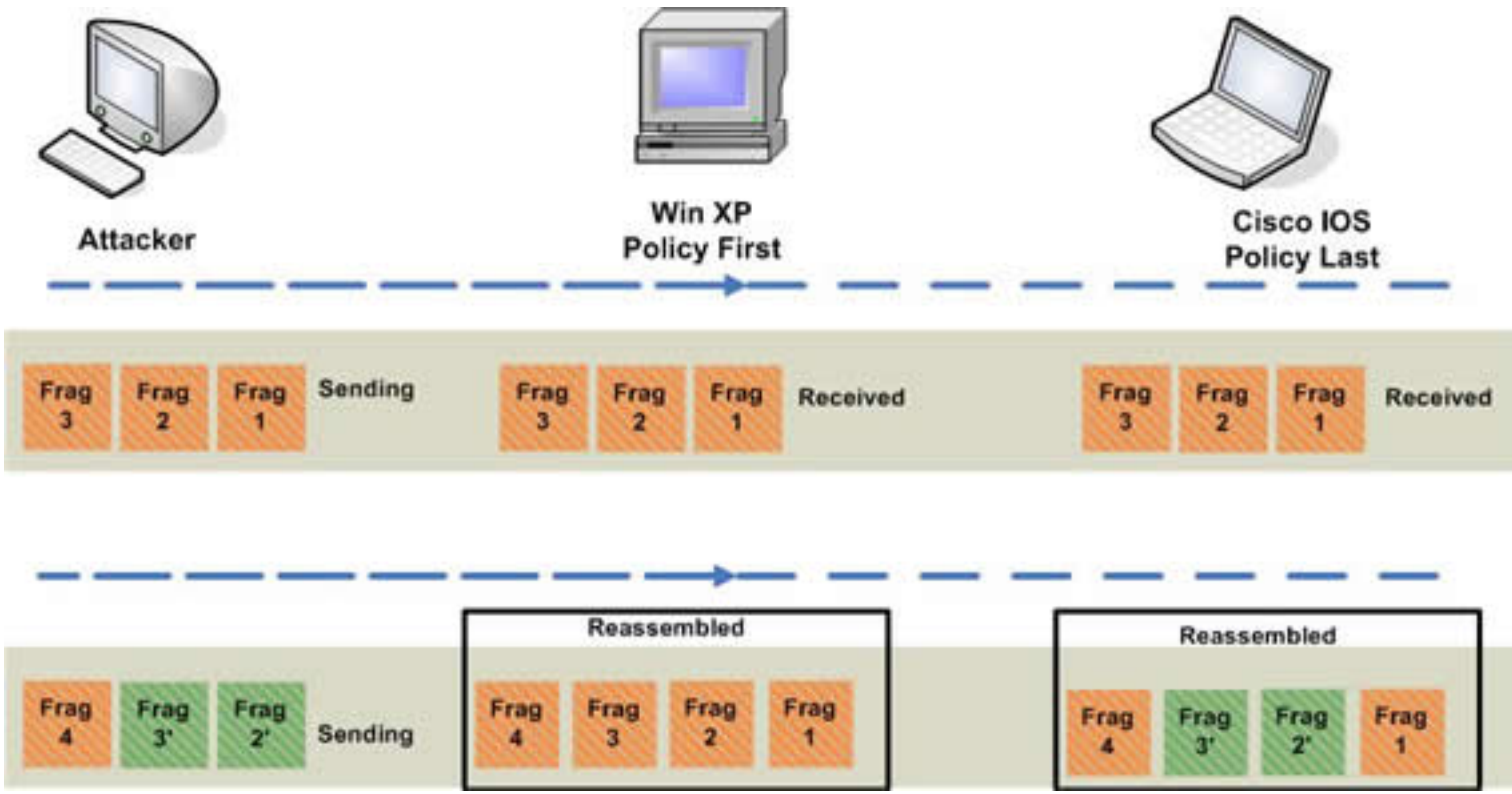
Evasion technique – Time-to-live

- Router drops packet analysed by NIDS that will not be delivered to victim



Evasion technique – Fragment replacement

- Some operating systems replace fragments with newer ones, others keep old fragments



Suggested reading

- Wool, Avishai. "A quantitative study of firewall configuration errors." *Computer* 37.6 (2004): 62-67.
- Axelsson, Stefan. "The base-rate fallacy and the difficulty of intrusion detection." *ACM Transactions on Information and System Security (TISSEC)* 3.3 (2000): 186-205.
- [Siddharth 2005] http://www.symantec.com/connect/articles/evading-nids-revisited