Overview

- Network model
- Secure configuration of devices
- Exchanging keys
- Secure networking protocols
- Firewalls
  - **Packet filter**
  - **Circuit-level gateway**
  - **Application-level gateway (aka proxy)**

2

Firewalls

- Firewall
  - **Located between local network and Internet**
    - ▸ **Intended as protection wall against attacks from outside**
    - ▸ **Controls all incoming and outgoing traffic**
      - ✓ Exclude all other access to local network
      - ✓ Possibility to generate alerts for detected anomalies (IDS functionality)
    - ▸ **Enforces restrictions on network traffic**
      - ✓ Only traffic authorised in security policy
    - ▸ **Must be itself immune from unauthorised access**
      - ✓ Requires reliable system / secure OS

3

Firewalls

- **Firewall controls**
  - **Control of services/direction**
    - Which Internet services (e-mail, WWW, etc.) are accessible
      - ✓ Internal services from outside
      - ✓ External services from within
  - **Control of users**
    - Access control to services, dependent on user, which requires authentication technique
  - **Behaviour control**
    - Control how services are used (e.g. spam filter, filter for some websites, protection against DoS, etc.)

4

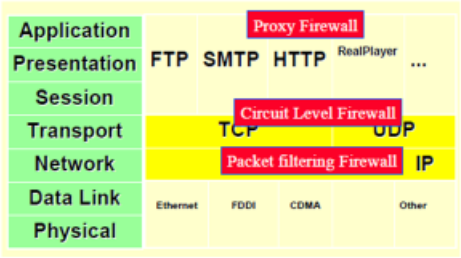
Firewalls

- **Limitations**
  - **Secures the *perimeter* of a company**
    - No protection against internal attacks or complicity from within
  - **No protection against attacks that circumvent the firewall**
    - Other access points to LAN may be vulnerable (wireless, mobile devices, etc.)
  - **Limited protection against malware**
    - Scanning all incoming traffic sometimes integrated in firewall, at significant computational cost

5

Firewalls

- A few types
  - Packet filter
  - Circuit-level gateway
  - Application-level gateway (aka proxy)



6

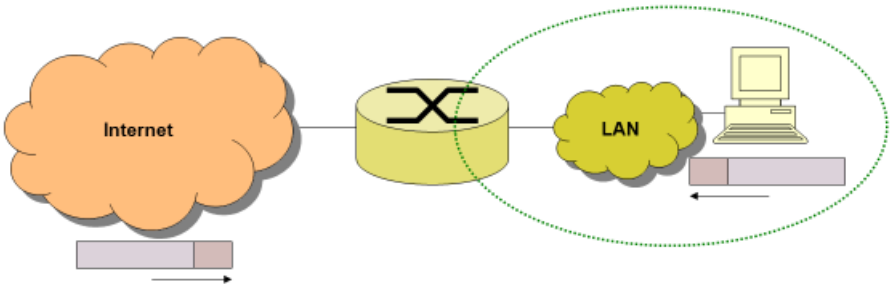
Overview

- Network model
- Secure configuration of devices
- Exchanging keys
- Secure networking protocols
- Firewalls
  - Packet filter
  - Circuit-level gateway
  - Application-level gateway (aka proxy)

7

Firewalls: packet filter

- **Packet filter**
  - **Simplest component**
    - Router analyses each incoming and outgoing IP packet
    - Decides, based on filter rules, which packets are accepted, and which ones are blocked



8

Packet filter

- **Filter rules based on information in IP packet**
  - **Source IP address**
  - **Destination IP address**
  - **(Source/destination) address at transport level**
    - TCP or UDP port (defines applications such as HTTP or SMTP)
  - **Transport protocol used**
  - **Router interface where packet arrives or is sent to**
- **Possible default policy for filter rules ("forward policy")**
  - **"Block"/"discard"**
    - What isn't explicitly authorised, is forbidden
      - ✓ More prudent…
      - ✓ …but less user friendly
      - ✓ Progressively setting up list of authorised services
  - **"Allow"/"forward"**
    - What isn't explicitly forbidden, is authorised

Creating filter rules for a packet filter has been compared to programming in assembler: you have unlimited freedom but at great implementation costs.

9

Packet filter

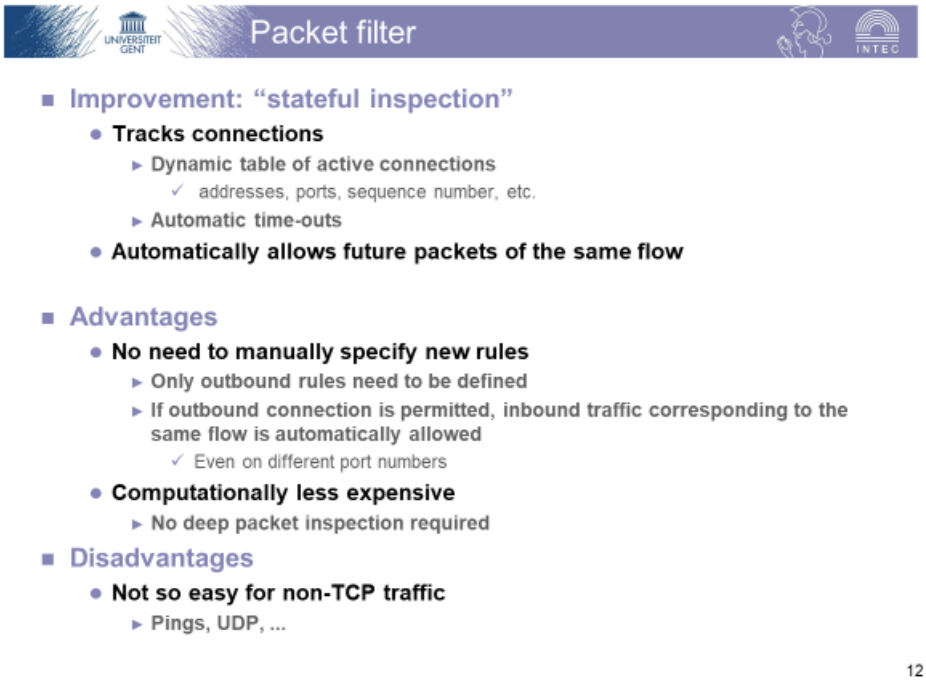
- Advantages
  - Basic security
  - Simple, fast, and relatively cheap
  - (Reasonably) transparent to applications and users
  - Non-cryptographic (and thus no key management)
- Drawbacks
  - No protection against attacks at the application level
    - All instructions for some application are authorised if the application is authorised
  - No (strong) user authentication
    - Unless combined with IPsec (AH protocol)
  - Configuration errors are quite common

10

Packet filter

- Issues
  - Risk of IP spoofing
    - Cause firewall to believe packet originates from authorised network
    - Can be averted by blocking incoming packets with an IP address from local network
      - ✓ Only possible if external networks can be denied access
    - Can be prevented using IPsec
  - Fragmentation attack
    - Split IP packet in very small fragments, causing TCP header information to be contained in next IP packet
    - To be prevented by blocking such tiny packets

11



Pure packet filters do not keep track of traffic and have no memory of previous packets which makes them vulnerable to spoofing attacks. Such a firewall has no way of knowing if any given packet is part of an existing connection, is trying to establish a new connection, or is just a rogue packet. In contrast, a stateful firewall (any firewall that performs stateful packet inspection (SPI) or stateful inspection) is a firewall that keeps track of the state of network connections (such as TCP streams, UDP communication) traveling across it. The firewall is programmed to distinguish legitimate packets for different types of connections. Only packets matching a known active connection will be allowed by the firewall; others will be rejected.

The classic example of a network operation that may fail with a stateless firewall is the File Transfer Protocol (FTP). By design, such protocols need to be able to open connections to arbitrary high ports to function properly. Since a stateless firewall has no way of knowing that the packet destined to the protected network (to some host's destination port 4970, for example) is part of a legitimate FTP session, it will drop the packet. Stateful firewalls with application inspection solve this problem by maintaining a table of open connections, inspecting the payload of some packets and intelligently associating new connection requests with existing legitimate connections. A stateful firewall keeps track of the state of network connections (such as TCP streams or UDP communication) and is able to hold significant attributes of each connection in memory. These attributes are collectively known as the state of the connection, and may include such details as the IP addresses and ports involved in the connection and the sequence numbers of the packets traversing the connection. Stateful inspection monitors incoming and outgoing packets over time, as well as the state of the connection, and stores the data in dynamic state tables. This cumulative data is evaluated, so that filtering decisions would not only be based on administrator-defined rules, but also on context that has been built by previous connections as well as previous packets belonging to the same connection.

Depending on the connection protocol, maintaining a connection's state is more or less complex for the firewall. For example, TCP is inherently a stateful protocol as connections are established with a three-way handshake ("SYN, SYN-ACK, ACK") and ended with a "FIN, ACK" exchange. This means that all packets with "SYN" in their header received by

the firewall are interpreted to open new connections. If the service requested by the client is available on the server, it will respond with a "SYN-ACK" packet which the firewall will also track. Once the firewall then receives the client's "ACK" response, it transfers the connection to the "ESTABLISHED" state as the connection has been authenticated bidirectionally. This allows tracking of future packets through the established connection. Simultaneously, the firewall drops all packets which are not associated with an existing connection recorded in its state table (or "SYN" packets), preventing unsolicited connections with the protected machine. Other connection protocols, namely UDP and ICMP, are not based on bidirectional connections like TCP, making a stateful firewall somewhat less secure. In order to track a connection state in these cases, a firewall must transfer sessions to the ESTABLISHED state after seeing the first valid packet. It can then only track the connection through addresses and ports of the following packets' source and destination. Unlike TCP connections, which can be closed by a "FIN, ACK" exchange, these connectionless protocols allow a session to end only by time-out.

By keeping track of the connection state, stateful firewalls provide added efficiency in terms of packet inspection. This is because for existing connections the firewall need only check the state table, instead of checking the packet against the firewall's rule set, which can be extensive. Additionally, in the case of a match with the state table, the firewall does not need to perform deep packet inspection.

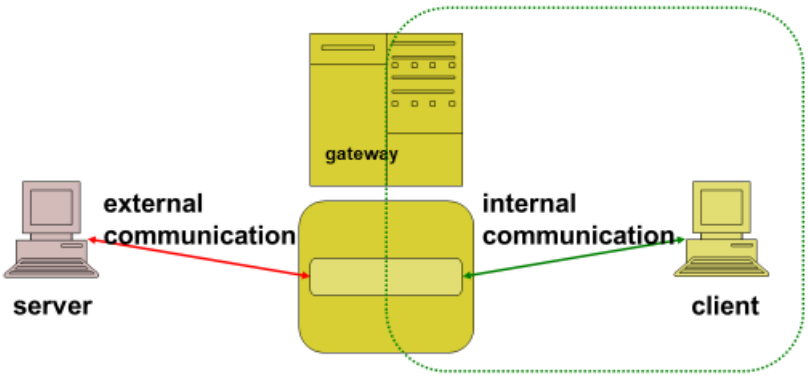
Overview

- Network model
- Secure configuration of devices
- Exchanging keys
- Secure networking protocols
- Firewalls
  - Packet filter
  - **Circuit-level gateway**
  - Application-level gateway (aka proxy)

13

Firewalls: circuit level gateway

- Circuit level gateway
  - **Operates as a relay at TCP level**
    - Or occasionally also at UDP level



14

A circuit level gateway can also be considered more or less as a proxy-server for TCP.

1) The gateway receives a request from the client to set up a TCP connection.
2) The gateway takes care of the client authentication and required authorisation
3) The gateway sets up a TCP connection with the server in name of the client

After this TCP connection setup the gateway will transmit the data from the internal TCP connection to the external TCP connection (and vice versa).

Circuit level gateway

- Advantages
  - Gateway doesn't need to know the application
  - Generic as concerns the applications used
  - Suitable to allow authenticated traffic through a firewall
  - Can be combined with proxy servers (see later)
    - Especially for applications for which there is no proxy server
- Drawbacks
  - Unable to intercept application specific threats
    - E.g. Java applets, ActiveX, SQL injection, etc.

15

Circuit level gateway

- SOCKS
  - Best known example of circuit level gateway
  - Requires a few modifications to client software or TCP/IP stack
    - In order to support SOCKS specific calls
    - Adding SOCKS-library to client software
    - Commonly used Web browsers are SOCKS compliant
  - Used for dynamic SSH port forwarding

16

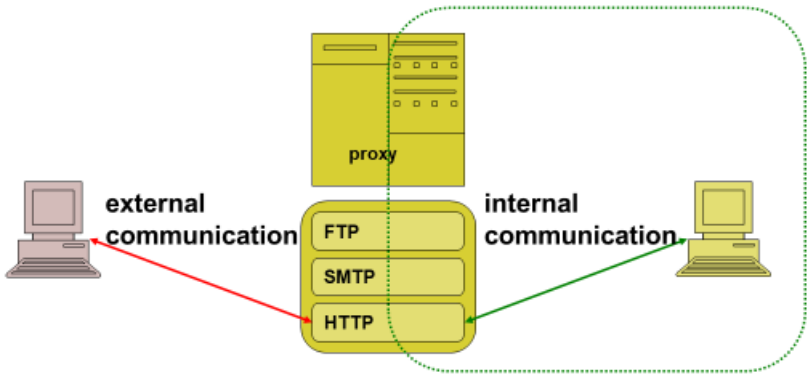
Overview

- Network model
- Secure configuration of devices
- Exchanging keys
- Secure networking protocols
- Firewalls
  - Packet filter
  - Circuit-level gateway
  - **Application-level gateway (aka proxy)**

17

Proxy

- Application level gateway aka proxy



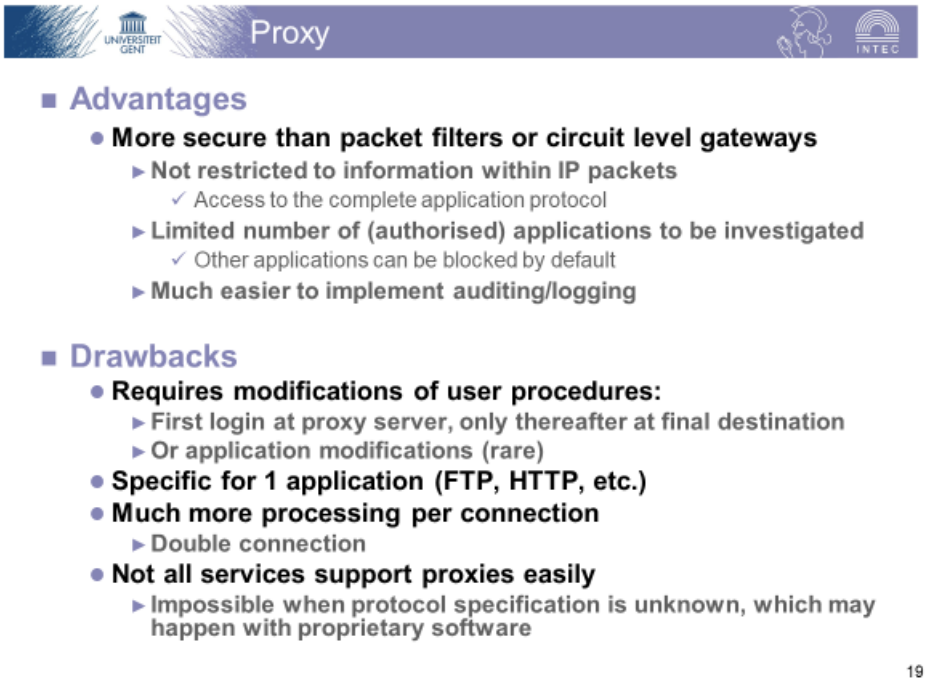
18

The third type of firewall is also called an application level gateway or "proxy". In contrast to previous options, a proxy is application specific and inspects application level packets (e.g. HTTP, SMTP, FTP, etc.). Traffic for these applications is only possible via the proxy. The proxy has access to the complete protocol specifics and can analyze all packet contents.

Sequence:

- The internal user requests a service from the proxy

- The proxy verifies and validates the request (and may also take care of client authentication)
    - Depending on the authentication, the proxy might not support certain parts of the service
    - The client authentication by the proxy can also be outsourced to an authentication server (e.g. RADIUS-server). This is even a desirable solution, as critical authentication information will the no longer be found at the firewall itself.
- The proxy forwards the request outward and returns the result to user



A simple example a possible application of a proxy server, for which a circuit level gateway or a packet filter would come short, is the following. Suppose that for an (anonymous) FTP-server (within the local network) one wants to allow both internal and external users to upload files on the server, but that only internal users may read these files. In this case a proxy-server may allow the PUT-instruction to pass the proxy, but will block the GET-instruction for all traffic that crosses the firewall perimeter. Another situation where a proxy may be advantageous is in blocking distributed software (e.g. web services), which is typically not blocked by a packet filter (TCP port 80 of HTTP).

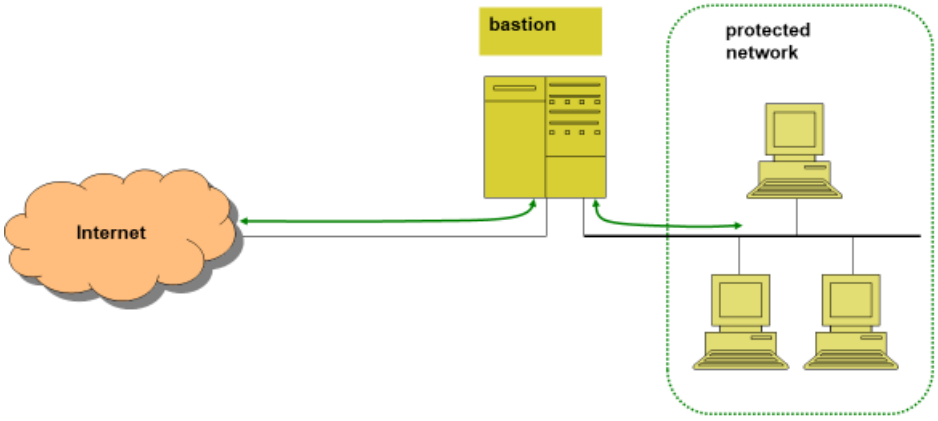
Firewalls: bastion

- **Bastion**
  - **Critical for security within network perimeter**
    - Exposed to external attacks
    - Typically used as gateway to the system
  - **Desirable properties**
    - Secure OS version
    - Only essential services are installed
      - ✓ Proxy applications (DNS, FTP, SMTP, user authentication, etc.)
      - ✓ Only minimally necessary set of instructions of applications is implemented
    - Possibility to use strong authentication for access to proxy services
  - **For sufficiently large systems**
    - Firewall is more than 1 single device (packet filter or gateway)
    - Combination of several elements
    - Some basic configurations in the following slides

20

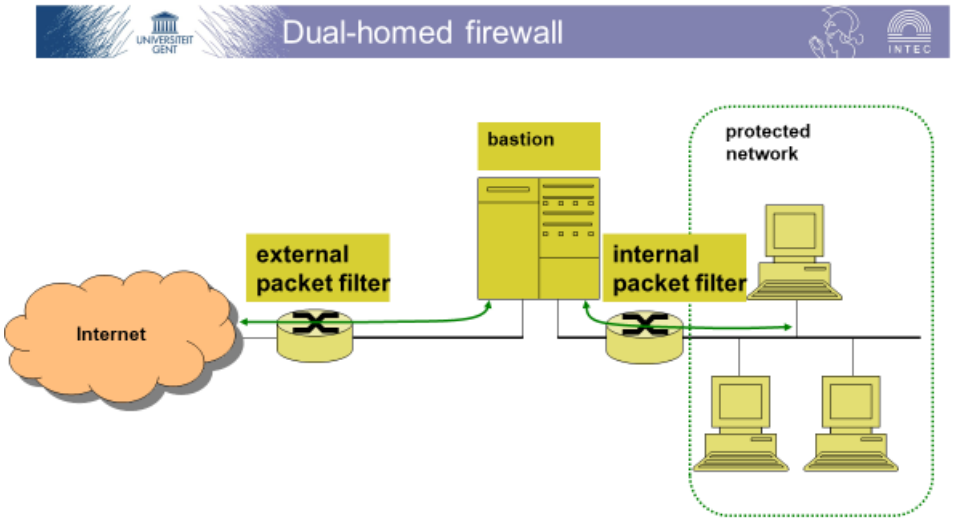
A Bastion host is a special purpose computer on a network specifically designed and configured to withstand attacks. The computer generally hosts a single application, for example a proxy server, and all other services are removed or limited to reduce the threat to the computer. It is hardened in this manner primarily due to its location and purpose, which is either on the outside of the firewall or in the DMZ and usually involves access from untrusted networks or computers.

Simple dual-homed firewall



21

A multi-homed host has more than 1 network interface, allowing to separate network parts, in this case between the external and the internal network.

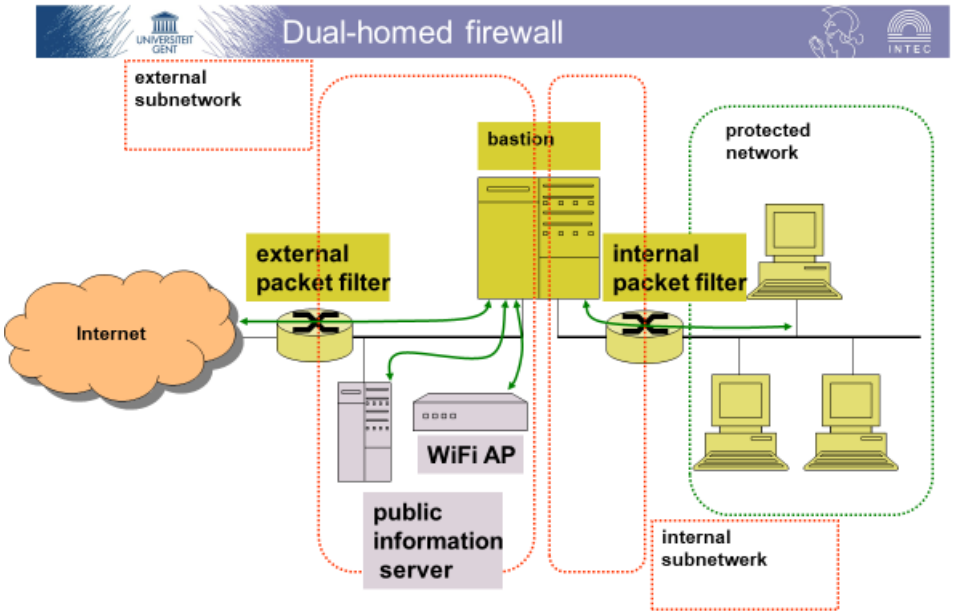


This is a more realistic approach of a dual-homed firewall. Three elements in this firewall system: external and internal packet filter, and bastion.

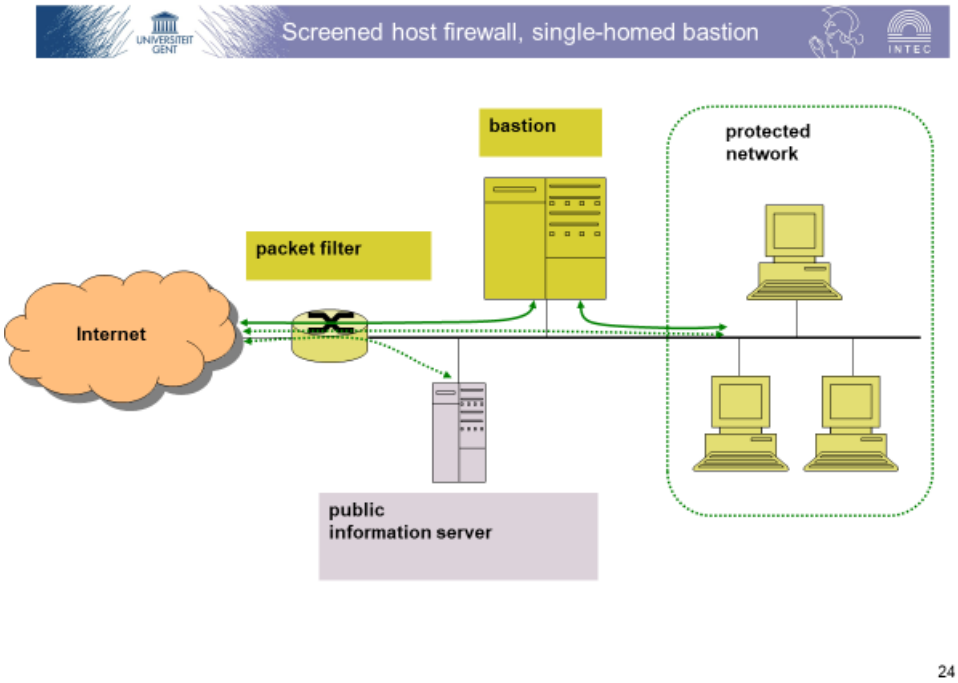
There is a (external) packet filter between the Internet and the bastion, and also a (internal) packet filter between the bastion and the protected network. The external packet filter only allows traffic between the Internet and the bastion (and no direct communication between the protected network and the Internet). The internal packet filter causes the protected network to be able to access the outside only through the bastion.

This configuration is simple and very secure, but not very flexible:

1) The bastion can be a limiting network performance factor as it has to handle all incoming and outgoing traffic.
2) If the bastion fails, connection to the outside is lost ("single point or failure").
3) Application protocols without proxy support are not allowed.



In this situation an external subnetwork and an internal subnetwork are created. The external subnetwork is typically adequate to host some non-critical services, such as a public webserver or access servers for other networks (e.g. wireless access points).

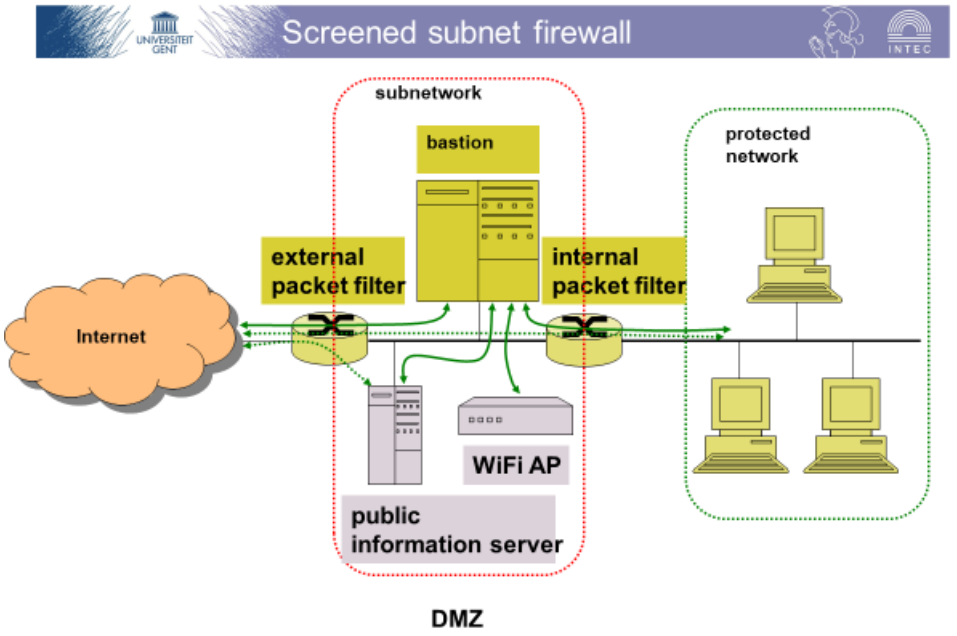


Two elements in firewall system: packet filter and bastion.

The packet filter is configured to allow only IP packets from the Internet to the bastion, and for packets originating from the protected network, only IP packets from the bastion to the outside.

The function of the bastion is to provide authentication and the proxy-functions.

For some (presumed secure) applications (e.g. the access to a public web server, but also IPsec traffic) it may be allowed to bypass the bastion and to authorise direct communication with some part of the protected network. The flexibility of this configuration is therefore much larger, but the degree of security is potentially lower (as it is possible to bypass the bastion for some part of the traffic).



Three elements in firewall system: external and internal packet filter, and bastion.

There is a (external) packet filter between the Internet and the bastion, and also a (internal) packet filter between the bastion and the protected network. An isolated subnetwork is thus created (which is screened by the bastion), which may also contain public information servers (this is also the zone where wireless access points are best located). Traffic from the Internet to the subnetwork and from the protected network to the subnetwork may be enabled, but any direct traffic between the Internet and the protected network will be blocked by the packet filters. The presence of the internal packet filter improves the protection compared to the situation in a "screened host firewall".

This is a configuration offering both a decent degree of protection and sufficient flexibility. Here too, it is possible to authorise for certain "secure" applications a direct communication between the protected network and the internet, without having to use the bastion).

The subnetwork is sometimes also called the "demilitarised zone" (in short DMZ).

Firewalls

- **Personal firewalls**
  - **Software on PC**
    - **Minimal perimeter (PC instead of complete network)**
    - **More limited functionality**
    - **Limited reliability**
      - ✓ Running on regular OS
    - **Still useful as basic protection against external trouble**

26

Firewalls

- **A few concluding remarks**
  - **Firewalls don't solve *all* security issues**
    - **Only effective if** all **incoming and outgoing traffic passes through the firewall**
      - ✓ Threat of wireless access points
    - **No protection against internal threats**
      - ✓ Including imported malware
  - **Protection of a local network as a fortified citadel is a somewhat medieval concept**
  - **The best protection won't withstand the creativity of internal users**

27