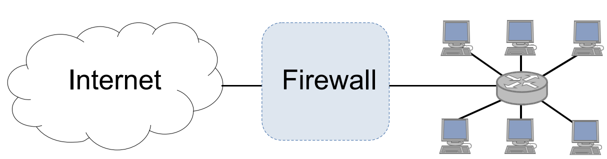
Protection of Communication Infrastructures

Chapter 5

Internet Firewalls

Introduction to Network Firewalls (1)

- In building construction, a firewall is designed to keep a fire from spreading from one part of the building to another
- A network firewall, however, can be better compared to a moat of a medieval castle:
  - It restricts people to entering at one carefully controlled point
  - It prevents attackers from getting close to other defenses
  - It restricts people to leaving at one carefully controlled point
- Usually, a network firewall is installed at a point where the protected subnetwork is connected to a less trusted network:
  - Example: Connection of a corporate local area network to the Internet



- So, basically firewalls realize access control on the subnetwork level

TELEMATIK Rechnernetze

Introduction to Network Firewalls (2)

- What firewalls can do:
  - A firewall is a focus for security decisions
  - A firewall can enforce a security policy, i.e. concerning access control
  - A firewall can log Internet activity efficiently
  - A firewall can limit exposure to security problems in one part of a network
- What firewalls can not do:
  - A firewall cannot protect against malicious insiders
  - A firewall cannot protect against connections that do not go through it
    - If, for example, there is an access point behind a firewall that provides unauthenticated access to the subnetwork, the firewall can not provide any protection against malicious WLAN users
  - A firewall cannot protect against completely new threats
  - A firewall cannot fully protect against viruses
  - A firewall cannot set itself up correctly (➡ cost of operation)

TELEMATIK Rechnernetze

Two Fundamental Approaches Regarding Firewall Policy

- Default deny strategy:
  - *"Everything that is not explicitly permitted is denied"*
  - Examine the services the users of the protected network need
  - Consider the security implications of these services and how the services can be safely provided
  - Allow only those services that can be safely provided and for which there is a legitimate need
  - Deny any other service
- Default permit strategy:
  - *"Everything that is not explicitly forbidden is allowed"*
  - Permit every service that is not considered dangerous
  - Example:
    - *Server Message Block (SMB)* and X-Windows is not permitted across the firewall
    - Incoming SSH connections are only allowed to one specific host

What Internet Services & Protocols are to be Considered?

- Electronic mail: Simple Mail Transfer Protocol (SMTP), IMAP, POP3
- File exchange: Web-based Distributed Authoring and Versioning (WebDAV), File Transfer Protocol (FTP), Network File System (NFS)
- Remote terminal access and command execution: Secure SHell (SSH)
- World wide web: HyperText Transfer Protocol (HTTP, HTTPS)
- Real-time conferencing services: ICQ, Jabber, Skype, Adobe Connect, ...
- Name services: Domain Name Service (DNS)
- Network management: Simple Network Management Protocol (SNMP)
- Time service: Network Time Protocol (NTP)
- Window systems: Remote Desktop Protocol (RDP), X-Windows
- Printing systems: Internet Printing Protocol (IPP)

Some Background on Internet Services, IP, TCP & UDP

- Internet services are usually realized with client and server programs and application protocols that are run by those programs
- The application protocol data units are most often transported in either segments of a TCP connection or UDP datagrams

Application Protocol	
TCP	UDP
IPv4 / IPv6	
Access Protocol	

- The TCP segments / UDP datagrams are transported in IP packets which themselves are transported in the PDUs of the data link technology used on the links between source and destination
  - Examples: Ethernet, WLAN, etc.
- The addressing of application processes (like clients, servers) is realized by the tuples:
  - Source IP address, source port
  - Destination IP address, destination port
  - A port is a two-byte number that identifies what application process the application PDU is coming from / going to

Example: An IPv4 Packet Carrying a TCP Segment

IP header	Ver.	IHL	ToS/DSCP	Length
	IP Identification		Flags	Fragment Offset
	TTL		Protocol	IP Checksum
	Source Address			
	Destination Address			
	IP Options (if any)			
TCP header	Source Port			Destination Port
	Sequence Number			
	Request Number			
	Offset	Reserv.	Control	Window
	Checksum			Urgent Pointer
	TCP Options (if any)			
Payload	Payload (Including application PDU header)			

Protocol Fields Important for Firewalls (1)

- Access Protocol:
  - Network Layer Protocol: IPv4, IPv6
  - Access Protocol Addresses: Ethernet MAC address, etc.
    - These addresses either refers to the final source / destination or the addresses of the intermediate nodes of this link
- IP:
  - Source address
  - Destination address
  - Flags, especially the indication of an IP fragment (in IPv6 an option)
  - Protocol type: TCP, UDP, ICMP, ...
  - Options:
    - Source routing:
      - the sender explicitly specifies the route an IP packet will take
      - as this is often used for attacks most firewalls discard these packets
    - In general, IP options are rarely used in IPv4

Protocol Fields Important for Firewalls (2)

- TCP:
  - Source port, Destination port:
    - Evaluation of source and destination ports allow to determine (with a limited degree of confidence) the sending / receiving application, as many Internet services use well-known port numbers
  - Control:
    - ACK: this bit is set in every segment but the very first one transmitted in a TCP connection, it therefore helps to identify connection requests
    - SYN: this bit is only set in the first two segments of a connection, so it can be used to identify connection confirmations
    - RST: if set this bit indicates an ungraceful close of a connection, it can be used to shut peers up without returning helpful error messages
- Application protocol:
  - In some cases a firewall might even need to peek into application protocol header fields
  - However, as this is application-dependent this class will not go into detail...

Firewall Terminology & Building Blocks for Firewalls (1)

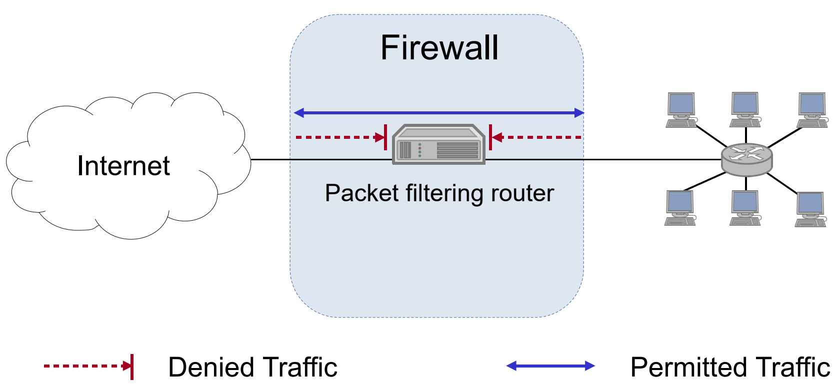
- *Firewall:*
  - A component or a set of components that restricts access between a protected network and the Internet or between other sets of networks
- *Packet filtering:*
  - The action a device takes to selectively control the flow of data to and from a network
  - Packet filtering is an important technique to implement access control on the subnetwork-level for packet oriented networks, e.g. the Internet
  - A synonym for packet filtering is *screening*
- *Bastion host:*
  - A computer that must be highly secured because it is more vulnerable to attacks than other hosts on a subnetwork
  - A bastion host in a firewall is usually the main point of contact for user processes of hosts of internal networks with processes of external hosts
- *Dual homed host:*
  - A general purpose computer with at least two network interfaces

Firewall Terminology & Building Blocks for Firewalls (2)

- *Proxy:*
  - A program that deals with external servers on behalf of internal clients
  - Proxies relay approved client requests to real servers and also relay the servers answers back to the clients
  - If a proxy interprets and understands the commands of an application protocol it is called an *application level proxy,* if it just passes the PDUs between the client and the server it is called a *circuit level proxy*
- *Network Address Translation (NAT):*
  - A procedure by which a router changes data in packets to modify the network addresses
  - This allows to conceal the internal network addresses (even though NAT is not actually a security technique)
- *Perimeter Network:*
  - A subnetwork added between an external and an internal network, in order to provide an additional layer of security
  - A synonym for perimeter network is *de-militarized zone (DMZ)*

Firewall Architectures (1)

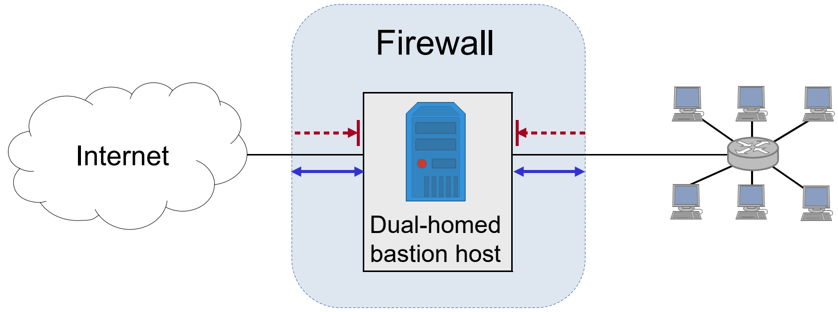
The Simple Packet Filter Architecture



- The most simple architecture just consists of a packet filtering router
- It can be either realized with:
  - A standard workstation (e.g. Linux PC) with at least two network interfaces plus routing and filtering software
  - A dedicated router device, which usually also offers filtering capabilities

Firewall Architectures (2)

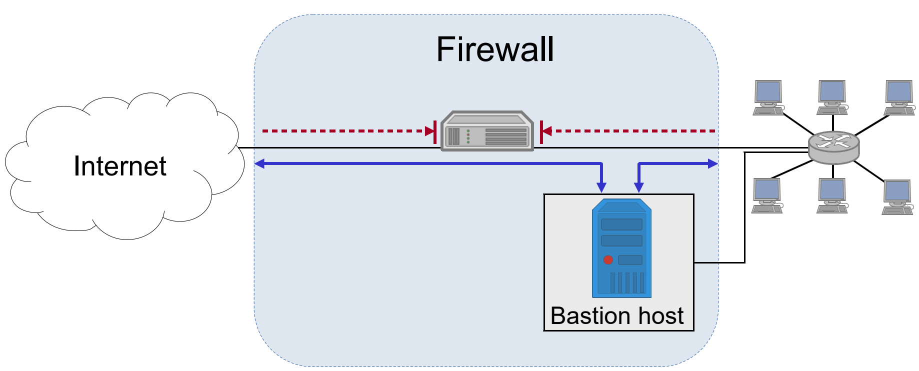
The Dual-Homed Host Architecture



- The dual-homed host provides:
  - Proxy services to internal and / or external clients
  - Potentially packet filtering capabilities if it is also acting as a router
- Properties of the dual-homed host:
  - It has at least two network interfaces
- Drawback: As all permitted traffic passes through the bastion host, this might introduce a performance bottleneck

Firewall Architectures (3)

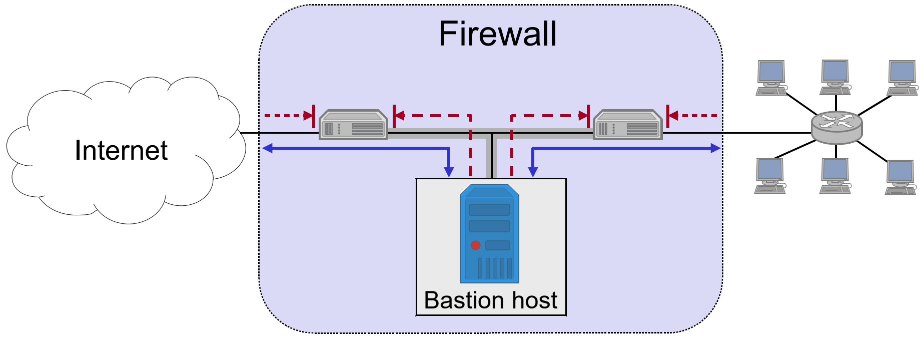
The Screened Host Architecture



- The packet filter:
  - Allows permitted IP traffic to flow between the screened host and the Internet
  - Blocks all direct traffic between other internal hosts and the Internet
- The screened host provides proxy services:
  - Despite partial protection by the packet filter the screened host acts as a bastion host

Firewall Architectures (4)

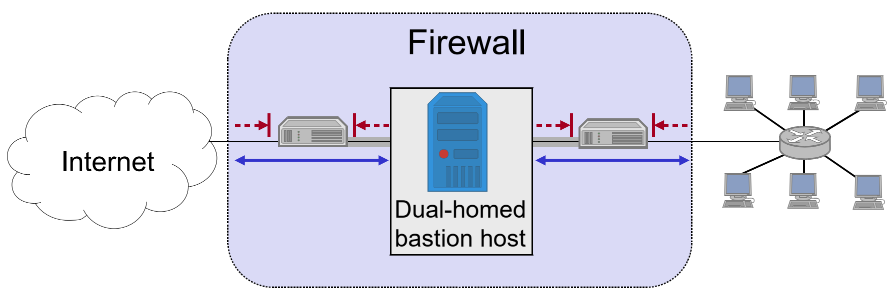
The Screened Subnet Architecture



- A perimeter network is created between two packet filters
- The inner packet filter serves for additional protection in case the bastion host is ever compromised:
  - For example, this avoids a compromised bastion host to sniff on internal traffic
- The perimeter network is also a good place to host a publicly accessible information server, e.g. a web server

Firewall Architectures (5)

The Split Screened Subnet Architecture



- A dual-homed bastion host splits the perimeter network in two distinct networks
- This provides defense in depth, as:
  - The dual-homed bastion host provides finer control on the connections as his proxy services are able to interpret application protocols
  - The bastion host is protected from external hosts by an outer packet filter
  - The internal hosts are protected from the bastion host by an inner packet filter

Packet Filtering (1)

- What can be done with packet filtering?
  - Theoretically speaking everything, as all information exchanged in a communication relation is transported via packets
  - In practice, however, the following observations serve as a guide:
    - Operations that require quite detailed knowledge of higher layer protocols or prolonged tracking of past events are easier to realize in proxy systems
    - Operations that are simple but need to be done fast and on individual packets are easier to do in packet filtering systems
- Basic packet filtering enables to control data transfer based on:
  - Source IP Address
  - Destination IP Address
  - Transport protocol
  - Source and destination application port
  - Potentially, specific protocol flags (e.g. TCP's ACK- and SYN-flag)
  - The network interface a packet has been received on

Packet Filtering (2)

- More elaborate packet filtering:
  - *Stateful or dynamic packet filtering:*
    - Example 1: *"Let incoming UDP packets through only if they are responses to outgoing UDP packets that have been observed"*
    - Example 2: *"Accept TCP packets with the SYN bit set only as part of TCP connection initiation"*
  - *Protocol checking:*
    - Example 1: *"Let in packets bound for the DNS port, but only if they are formatted like DNS packets"*
    - Example 2: *"Do not allow HTTP transfers to these sites"*
  - However, more elaborate packet filtering consumes more resources!
- Actions of a packet filter:
  - Pass the packet
  - Drop the packet
  - Potentially, log the passed or dropped packet (entirely or parts of it)
  - Potentially, pass an error message to the sender (may help an attacker!)

Packet Filtering (3)

- Specifying packet filtering rules:
  - As a packet filter protects one part of a network from another one, there is an implicit notion of the direction of traffic flow:
    - *Inbound:* The traffic is coming from an interface which is outside the protected network and its destination can be reached on an interface which is connected to the protected network
    - *Outbound:* the opposite of inbound
    - For every packet filtering rule this direction is specified as either *"inbound"*, *"outbound"*, or *"either"*
  - Source and destination address specifications can make use of wildcards, e.g. 125.26.0.0/16 denotes all addresses starting with 125.26.
    - In our examples, we denote often simply denote addresses as *"internal"* or *"external"* when we want to leave exact network topology out of account
  - For source and destination ports we sometimes write ranges, e.g. ">1023"
  - We assume filtering rules to be applied in the order of specification, that means the first rule that matches a packet is applied

An Example Packet Filtering Ruleset (1)

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest. Port	ACK	Action
A	Inbound	External	Internal	TCP		25		Permit
B	Outbound	Internal	External	TCP		>1023		Permit
C	Outbound	Internal	External	TCP		25		Permit
D	Inbound	External	Internal	TCP		>1023		Permit
E	Either	Any	Any	Any		Any		Deny

- This first ruleset aims to specify, that incoming and outgoing email should be the only allowed traffic into and out of a protected network
- Email is relayed between two servers by transferring it to an SMTP-daemon on the target server (server port 25, client port > 1023)
- Rule A allows incoming email to enter the network and rule B allows the acknowledgements to exit the network
- Rules C and D are analogous for outgoing email
- Rule E denies all other traffic

An Example Packet Filtering Ruleset (2)

- Consider, for example, a packet which “wants” to enter the protected subnet and has a forged IP source address from the internal network:
  - As all allowed inbound packets must have external source and internal destination addresses (A, D) this packet is successfully blocked
  - The same holds for outbound packets with external source addresses (B, C)
- Consider now SSH traffic:
  - As a telnet server resides usually at port 22, and all allowed inbound traffic must be either to port 25 or to a port number > 1023, incoming packets to initiate an incoming telnet connection are successfully blocked
  - The same holds for outgoing SSH connections
- However, the ruleset is flawed as, for example, it does not block the RDP-protocol for terminal server applications:
  - An RDP server usually listens at port 3389, clients use port numbers > 1023
  - Thus, an incoming RDP request is not blocked (B), neither is any answer (D)
  - This is highly undesirable, as the RDP protocol may allow attackers to log into clients with weak passwords

An Example Packet Filtering Ruleset (3)

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest. Port	ACK	Action
A	Inbound	External	Internal	TCP	>1023	25		Permit
B	Outbound	Internal	External	TCP	25	>1023		Permit
C	Outbound	Internal	External	TCP	>1023	25		Permit
D	Inbound	External	Internal	TCP	25	>1023		Permit
E	Either	Any	Any	Any	Any	Any		Deny

- The above flaw can be fixed by including the source ports into the ruleset specification:
  - Now outbound traffic to ports >1023 is allowed only if the source port is 25 (B), traffic from internal RDP clients or servers (port >1023) will be blocked
  - The same holds for inbound traffic to ports >1023 (D)
- However, it can not be assumed for sure, that an attacker will not use port 25 for his attacking RDP client:
  - In this case the above filter will let the traffic pass

An Example Packet Filtering Ruleset (4)

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest. Port	ACK	Action
A	Inbound	External	Internal	TCP	>1023	25	Any	Permit
B	Outbound	Internal	External	TCP	25	>1023	Yes	Permit
C	Outbound	Internal	External	TCP	>1023	25	Any	Permit
D	Inbound	External	Internal	TCP	25	>1023	Yes	Permit
E	Either	Any	Any	Any	Any	Any	Any	Deny

- This problem can be addressed by also specifying TCP's ACK-bit in rules B and D:
  - As the ACK-bit is required to be set in rule B, it is not possible to open a new TCP connection in the outbound direction to ports >1023, as TCP's connect-request is signaled with the ACK-bit not set
  - The same holds for the inbound direction, as rule D requires the ACK bit to be set
- As a basic rule, any filtering rule that permits incoming TCP packets for outgoing connections should require the ACK-bit be set

An Example Packet Filtering Ruleset (5)

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest. Port	ACK	Action
A	Inbound	External	Bastion	TCP	>1023	25	Any	Permit
B	Outbound	Bastion	External	TCP	25	>1023	Yes	Permit
C	Outbound	Bastion	External	TCP	>1023	25	Any	Permit
D	Inbound	External	Bastion	TCP	25	>1023	Yes	Permit
E	Either	Any	Any	Any	Any	Any	Any	Deny

- If the firewall comprises a bastion host, the packet filtering rules should further restrict traffic flow (→ screened host architecture):
  - As in the modified rules above only traffic between the Internet and the bastion host is allowed, external attackers can not attack SMTP on arbitrary internal hosts any longer
- In a screened subnet firewall, two packet filtering routers are set up:
  - one for traffic allowed between the Internet and the bastion host, and
  - one for traffic allowed between the bastion host and the internal network

Bastion Hosts (1)

- A bastion host is defined as a host that is more exposed to the hosts of an external network than the other hosts of the network it protects
- A bastion host may serve for different purposes:
  - Packet filtering
  - Providing proxy services
  - A combination of both
- The principles for building a bastion hosts are extensions of those for securing any mission critical host:
  - Keep it simple
  - Prepare for the bastion host to be compromised:
    - Internal hosts should not trust it any more than is absolutely required
    - If possible, it should be connected in a way to the network so that it can not sniff on internal traffic
    - Provide extensive logging for incident detection / analysis, if possible such that it can not be easily tampered with even when the host is compromised

Bastion Hosts (2)

- Further guidelines:
  - Make the bastion host unattractive:
    - The fewer tools are available on the bastion host, the less useful the machine is to an attacker
  - Get a reliable hardware configuration (no leading / bleeding edge)
  - The bastion host should be placed at a physically secure location
  - Disable all user accounts on the bastion host
  - Use different passwords (or with public key authentication none at all)
  - Secure the system logs (by writing them directly to a system which is not networked)
  - Do regular backups of the system logs and the configuration (using a dedicated backup device)
  - Monitor the machine closely (reboots, usage / load patterns, etc.)
  - If possible, restore the machine regularly from a prepared installation

Proxy Services (I)

- Proxying provides access to a specific Internet service for a single host, while appearing to provide it for all hosts of a protected network
- Candidate services for proxying:
  - FTP, Telnet, DNS, SMTP, HTTP
- Proxy servers usually run on (possibly dual-homed) bastion hosts
- The use of a proxy service usually leads to the following situation:
  - The user of a proxy service has the illusion of exchanging data with the actual server host
  - The actual server has the illusion of exchanging data with the proxy host

Proxy Services (II)

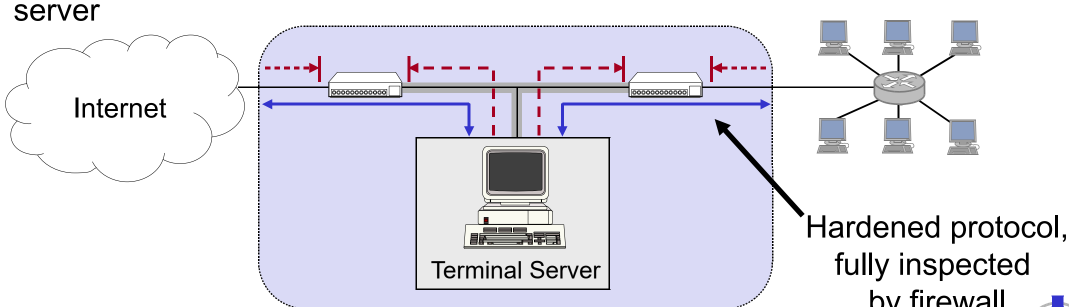
- Types of proxies
  - Application Level Proxy:
    - "Understands" application semantics
    - May scan for viruses, filter ads, cache content …
  - Circuit Level Proxy:
    - Forwards application PDUs without change
    - Usually only deployed if there is no specific Application Level Proxy, e.g. games
    - Most prominent example: SOCKS
    - Loosing significance due to NAT

Proxy Services (III)

- In order to instruct the proxy service to which server it should connect to, one of the following approaches can be taken:
  - Proxy-aware user procedures: Users log manually into an intermediate system
  - Proxy-aware client software: Users add proxy address to client software
  - Proxy-aware operating system: Proxy addresses are deployed by DHCP and Web Proxy Autodiscovery Protocol (WPAD)
  - Proxy-aware router: Routers intercept traffic and redirect to proxy server (*Transparent Proxy*)

Aspects of Modern Firewall Systems (I)

- One-Way-Gateways or Data Diodes are specialized firewalls that allow traffic only in one direction
  - Hardware-based enforcement, e.g., fiber only in one direction
  - Implemented as proxies as protocols usually assume bidirectional flows
  - To separate highly confidential networks (e.g. police or military) or networks of high integrity (e.g. to monitor nuclear power plants)
- Remote-Controlled Browsers System (ReCoBS) [Bun08]
  - Problem: Browsers are targets of attacks
  - Solution: Let users surf in a controlled environment on a hardened terminal server



Aspects of Modern Firewall Systems (II)

- Deep Packet Inspection, SSL Inspection and SSH Inspection
  - Modern firewalls analyze protocol behavior up to application layer
  - Some even JavaScript applications…
  - Problem: Encryption in HTTPS, SSH etc.
  - “Solution”
    - Automatic man-in-the-middle attacks
    - Using certificates from a locally trusted CA

- Network Access Control (NAC) and Unified Threat Management (UTM)
  - Modern firewalls are often tightly integrated in
    - User and device management (NAC)
    - Intrusion detection
    - Antivirus scanning
    - Network monitoring
  - It is UTM if it comes all from a single (potentially cheap) box

Additional References

[Bun08] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATI- ONSTECHNIK: Common Criteria Protection Profile for Remote-Controlled Browsers Systems (ReCoBS). BSI PP BSI-PP-0040. Version:2008

[Sem96a] C. Semeria. *Internet Firewalls and Security.* 3Com Technical Paper, 1996.

[SH09] SCARFONE, K. ; HOFFMAN, P.: Guidelines on Firewalls and Firewall Policy. NIST Special Publication 800-41. Version: 2009

[Wack95a] J. P. Wack, L.J. Carnahan. *Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls.* NIST Special Publication 800-10, 1995.

[Zwi00a] E. Zwicky, S. Cooper, B. Chapman. *Building Internet Firewalls.* Second Edition, O’Reilly, 2000.