Technische Hochschule Ingolstadt
Fakultät Informatik

Kapitel 3: Sicherheitsarchitekturen

CASE_SMN WS 2023/2024

Vorlesung „Sicherheit moderner Netze“

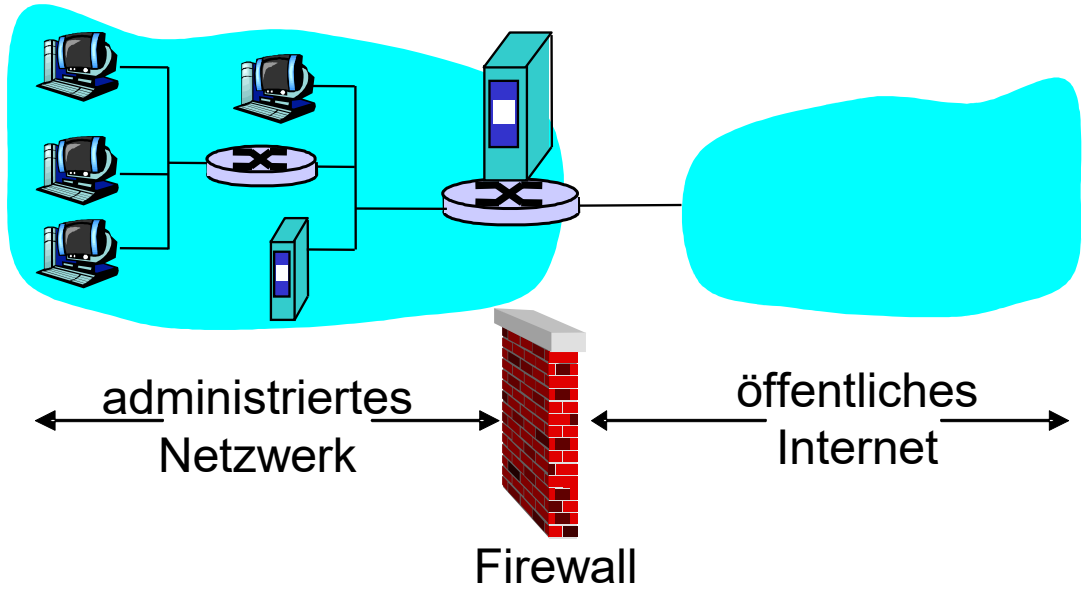
13.11.2023

- Mit welcher **Architektur** muss ein Netzbetreiber

(Campus, Enterprise, LAN)

sich gegen Angriffe aus dem Netz schützen?

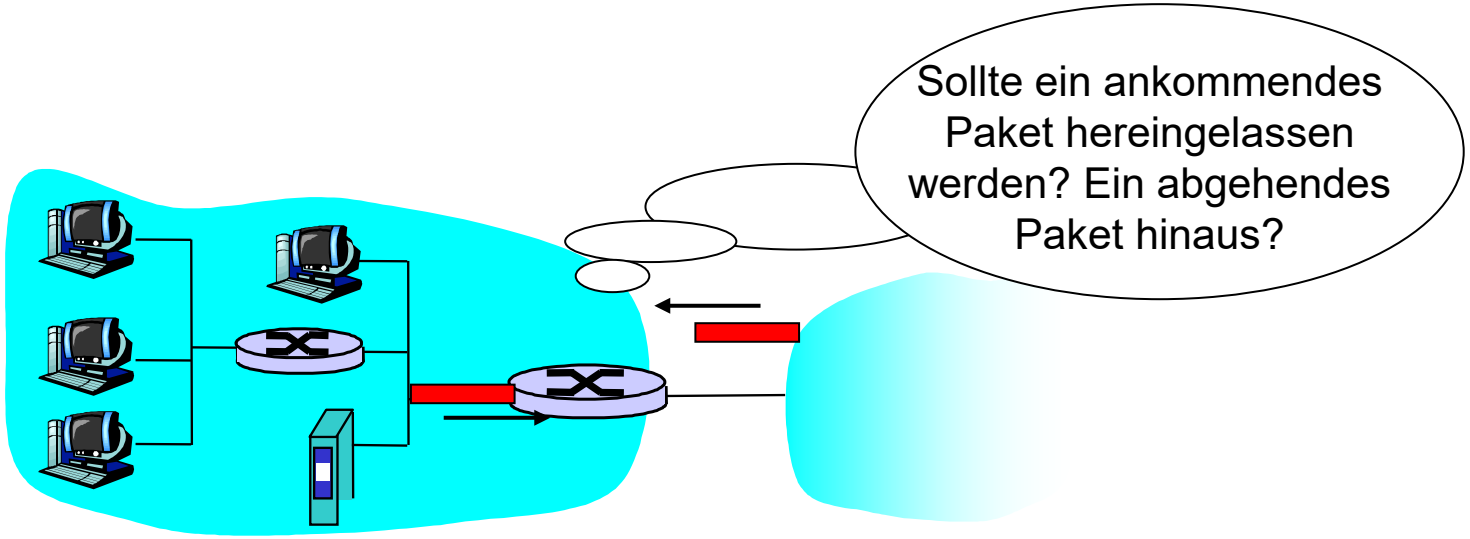
- **Eine Firewall trennt das interne Netz der Organisation vom Rest des Internet; manche Pakete dürfen passieren, andere werden herausgefiltert.**
  - Unterbinden von ungewolltem Datenverkehr von externen Computersystemen (WAN) zum geschützten Bereich (LAN)
  - Unterbinden von ungewolltem Datenverkehr vom LAN zum Internet



Aufgaben einer Firewall

- **Denial-of-Service-Angriffe abwehren:**
  - SYN-Flooding: Angreifer baut viele nutzlose TCP-Verbindungen auf, es bleiben keine Ressourcen für die “richtigen” Verbindungen
- **Illegalen Zugriff auf oder Manipulation von internen Daten verhindern:**
  - ein Angreifer könnte z.B. die Firmen-Homepage durch eine eigene Version ersetzen
- **Nur autorisierten Zugriff auf das interne Netz erlauben (definierte Menge von autorisierten Hosts/Benutzern)**
- **Drei Arten von Firewalls:**
  - zustandslose Paketfilter
  - zustandsbasierte Paketfilter
  - Anwendungs-Gateways

Zustandslose Paketfilter



- **internes Netz ist mit dem Internet über eine Router-Firewall verbunden**
- **Der Router betrachtet jedes Paket für sich, die Entscheidung, ob weitergeleitet wird, basiert auf :**
  - Quell- und Ziel-IP
  - TCP/UDP-Quell- und Zielportnummern
  - ICMP-Nachrichtentyp
  - TCP-SYN- und ACK-Bits

Zustandslose Paketfilter – Beispiel

- **Beispiel 1:**

Blockiere eingehende und ausgehende Datagramme mit

IP-Protokollfeld 17 und

entweder Quell- oder Ziel-Port 23.

  - alle ein- oder ausgehenden UDP-Flows und Telnet-Verbindungen werden blockiert.

- **Beispiel 2:**

Eingehende TCP-Segmente mit ACK=0 blockieren.

  - hält externe Hosts davon ab, zu internen Hosts TCP-Verbindungen aufzubauen, erlaubt es aber internen Hosts, nach Verbindungen nach außen zu initiieren.

Zustandslose Paketfilter – Beispiel

Ziel	Firewall-Regel
Kein Web-Zugriff nach außen.	
Keine eingehenden TCP-Verbindungen, außer sie sprechen den eigenen Webserver an.	
Vermeiden, dass Web-Radio die gesamte Bandbreite belegt.	
Verhindern, dass das eigene Netzwerk für einen DoS-Angriff missbraucht wird.	
Verhindern, dass das eigene Netz mit Traceroute untersucht wird.	

Zustandslose Paketfilter – Beispiel

Ziel	Firewall-Regel
Kein Web-Zugriff nach außen.	Alle eingehenden Pakete zu jeder IP-Adresse und Port 80 verwerfen.
Keine eingehenden TCP-Verbindungen, außer sie sprechen den eigenen Webserver an.	
Vermeiden, dass Web-Radio die gesamte Bandbreite belegt.	
Verhindern, dass das eigene Netzwerk für einen DoS-Angriff missbraucht wird.	
Verhindern, dass das eigene Netz mit Traceroute untersucht wird.	

Zustandslose Paketfilter – Beispiel (130.207.x.x)

Ziel	Firewall-Regel
Kein Web-Zugriff nach außen.	Alle eingehenden Pakete zu jeder IP-Adresse und Port 80 verwerfen.
Keine eingehenden TCP-Verbindungen, außer sie sprechen den eigenen Webserver an.	Alle eingehenden TCP-SYN-Pakete verwerfen, außer sie gehen an IP-Adresse 130.207.244.203, Port 80.
Vermeiden, dass Web-Radio die gesamte Bandbreite belegt.	
Verhindern, dass das eigene Netzwerk für einen DoS-Angriff missbraucht wird.	
Verhindern, dass das eigene Netz mit Traceroute untersucht wird.	

Zustandslose Paketfilter – Beispiel (130.207.x.x)

Ziel	Firewall-Regel
Kein Web-Zugriff nach außen.	Alle eingehenden Pakete zu jeder IP-Adresse und Port 80 verwerfen.
Keine eingehenden TCP-Verbindungen, außer sie sprechen den eigenen Webserver an.	Alle eingehenden TCP-SYN-Pakete verwerfen, außer sie gehen an IP-Adresse 130.207.244.203, Port 80.
Vermeiden, dass Web-Radio die gesamte Bandbreite belegt.	Alle eingehenden UDP-Pakete verwerfen, außer DNS und Router-Broadcasts.
Verhindern, dass das eigene Netzwerk für einen DoS-Angriff missbraucht wird.	
Verhindern, dass das eigene Netz mit Traceroute untersucht wird.	

Zustandslose Paketfilter – Beispiel (130.207.x.x)

Ziel	Firewall-Regel
Kein Web-Zugriff nach außen.	Alle eingehenden Pakete zu jeder IP-Adresse und Port 80 verwerfen.
Keine eingehenden TCP-Verbindungen, außer sie sprechen den eigenen Webserver an.	Alle eingehenden TCP-SYN-Pakete verwerfen, außer sie gehen an IP-Adresse 130.207.244.203, Port 80.
Vermeiden, dass Web-Radio die gesamte Bandbreite belegt.	Alle eingehenden UDP-Pakete verwerfen, außer DNS und Router-Broadcasts.
Verhindern, dass das eigene Netzwerk für einen DoS-Angriff missbraucht wird.	Eingehende ICMP-Pakete mit Broadcast-Zieladresse (z.B. 130.207.255.255) verwerfen.
Verhindern, dass das eigene Netz mit Traceroute untersucht wird.	

Zustandslose Paketfilter – Beispiel (130.207.x.x)

Ziel	Firewall-Regel
Kein Web-Zugriff nach außen.	Alle eingehenden Pakete zu jeder IP-Adresse und Port 80 verwerfen.
Keine eingehenden TCP-Verbindungen, außer sie sprechen den eigenen Webserver an.	Alle eingehenden TCP-SYN-Pakete verwerfen, außer sie gehen an IP-Adresse 130.207.244.203: Port 80.
Vermeiden, dass Web-Radio die gesamte Bandbreite belegt.	Alle eingehenden UDP-Pakete verwerfen, außer DNS und Router-Broadcasts.
Verhindern, dass das eigene Netzwerk für einen DoS-Angriff missbraucht wird.	Eingehende ICMP-Pakete mit Broadcast-Zieladresse (z.B. 130.207.255.255) verwerfen.
Verhindern, dass das eigene Netz mit Traceroute untersucht wird.	Ausgehende ICMP-TTL-Expired-Pakete verwerfen.

Access Control Lists (ACL)

- **ACL: Liste von Regeln, die von oben nach unten auf eingehende Pakete angewandt wird: Definiert die Kriterien und Aktionen (Bsp. Host im Netz 222.222 /16)**
- **Bsp.: Erlaube nur Webzugriff und DNS (Port 53)**

Aktion	Quell-IP	Ziel-IP	Protokoll	Quell-Port	Ziel-Port	Flags
erlaube	nicht in 222.222/16	222.222/16	TCP			
erlaube	222.222/16	nicht in 222.222/16	TCP			
erlaube	nicht in 222.222/16	222.222/16	UDP			
erlaube	222.222/16	nicht in 222.222/16	UDP			

Access Control Lists (ACL)

- **ACL: Liste von Regeln, die von oben nach unten auf eingehende Pakete angewandt wird: Definiert die Kriterien und Aktionen (Bsp. Host im Netz 222.222 /16)**
- **Bsp.: Erlaube nur Webzugriff und DNS (Port 53)**

Aktion	Quell-IP	Ziel-IP	Protokoll	Quell-Port	Ziel-Port	Flags
erlaube	nicht in 222.222/16	222.222/16	TCP	> 1023	80	any
erlaube	222.222/16	nicht in 222.222/16	TCP			
erlaube	nicht in 222.222/16	222.222/16	UDP			
erlaube	222.222/16	nicht in 222.222/16	UDP			

Access Control Lists (ACL)

- **ACL: Liste von Regeln, die von oben nach unten auf eingehende Pakete angewandt wird: Definiert die Kriterien und Aktionen (Bsp. Host im Netz 222.222 /16)**
- **Bsp.: Erlaube nur Webzugriff und DNS (Port 53)**

Aktion	Quell-IP	Ziel-IP	Protokoll	Quell-Port	Ziel-Port	Flags
erlaube	nicht in 222.222/16	222.222/16	TCP	> 1023	80	any
erlaube	222.222/16	nicht in 222.222/16	TCP	80	> 1023	ACK
erlaube	nicht in 222.222/16	222.222/16	UDP			
erlaube	222.222/16	nicht in 222.222/16	UDP			

Access Control Lists (ACL)

- **ACL: Liste von Regeln, die von oben nach unten auf eingehende Pakete angewandt wird: Definiert die Kriterien und Aktionen (Bsp. Host im Netz 222.222 /16)**
- **Bsp.: Erlaube nur Webzugriff und DNS (Port 53)**

Aktion	Quell-IP	Ziel-IP	Protokoll	Quell-Port	Ziel-Port	Flags
erlaube	nicht in 222.222/16	222.222/16	TCP	> 1023	80	any
erlaube	222.222/16	nicht in 222.222/16	TCP	80	> 1023	ACK
erlaube	nicht in 222.222/16	222.222/16	UDP	> 1023	53	---
erlaube	222.222/16	nicht in 222.222/16	UDP			

Access Control Lists (ACL)

- **ACL: Liste von Regeln, die von oben nach unten auf eingehende Pakete angewandt wird: Definiert die Kriterien und Aktionen (Bsp. Host im Netz 222.222 /16)**
- **Bsp.: Erlaube nur Webzugriff und DNS (Port 53)**

Aktion	Quell-IP	Ziel-IP	Protokoll	Quell-Port	Ziel-Port	Flags
erlaube	nicht in 222.222/16	222.222/16	TCP	> 1023	80	any
erlaube	222.222/16	nicht in 222.222/16	TCP	80	> 1023	ACK
erlaube	nicht in 222.222/16	222.222/16	UDP	> 1023	53	---
erlaube	222.222/16	nicht in 222.222/16	UDP	53	> 1023	----

Access Control Lists (ACL)

- **ACL: Liste von Regeln, die von oben nach unten auf eingehende Pakete angewandt wird: Definiert die Kriterien und Aktionen (Bsp. Host im Netz 222.222 /16)**
- **Bsp.: Erlaube nur Webzugriff und DNS (Port 53)**

Aktion	Quell-IP	Ziel-IP	Protokoll	Quell-Port	Ziel-Port	Flags
erlaube	nicht in 222.222/16	222.222/16	TCP	> 1023	80	any
erlaube	222.222/16	nicht in 222.222/16	TCP	80	> 1023	ACK
erlaube	nicht in 222.222/16	222.222/16	UDP	> 1023	53	---
erlaube	222.222/16	nicht in 222.222/16	UDP	53	> 1023	----
verbiete	alle	alle	alle	alle	alle	alle

Zustandsbehaftete Paketfilter

- **zustandslose Paketfilter sind oft unbeholfen:**
  - Pakete werden zugelassen, die “keinen Sinn machen”,

  z.B. Ziel-Port 80, ACK-Flag gesetzt, obwohl keine TCP-Verbindung existiert:

Aktion	Quell-IP	Ziel-IP	Protokoll	Quell-Port	Ziel-Port	Flags
erlaube	nicht in 222.222/16	222.222/16	TCP	80	> 1023	ACK

- **zustandsbehafteter Paketfilter: verfolgt den Zustand jeder TCP-Verbindung**
  - liest den Verbindungsauf- (SYN) und -abbau (FIN) mit: kann daher bestimmen, ob ein- und ausgehende Pakete “sinnvoll” sind
  - Timeout für inaktive Verbindungen in der Firewall: alte Verbindungen nicht mehr durchlassen

Zustandsbehaftete Paketfilter

- **ACL wird erweitert um anzuzeigen, ob es notwendig ist, die Zustandstabelle der Verbindungen ebenfalls zu prüfen Eigenes Netz: 222.222 /16**

Action	Source Address	Destination Address	Prot	Source Port	Dest Port	Flag Bit	Check connection
erlaube	nicht in 222.222/16	222.222/16	TCP	> 1023	80	any	
erlaube	222.222/16	nicht in 222.222/16	TCP	80	> 1023	ACK	X
erlaube	nicht in 222.222/16	222.222/16	UDP	> 1023	53	---	
erlaube	222.222/16	nicht in 222.222/16	UDP	53	> 1023	----	X
Verbiete	all	all	all	all	all	all	

Transport	Port	Protokoll	Beschreibung	gesperrte Richtung
UDP	67	bootps	bootp/DHCP Server	von aussen
UDP	68	bootpc	bootp/DHCP Client	von aussen
UDP	123	ntpd	Time Service	von aussen
UDP, TCP	135	loc-srv	MS DCE Locator Service/Endpoint Mapper	beide
UDP, TCP	137	NeTBIOS	NETBIOS Name Service	beide
UDP, TCP	138	NeTBIOS	NETBIOS Datagram Service	beide
UDP, TCP	139	NeTBIOS	NETBIOS Session Service	beide
UDP, TCP	161	SNMP	Netzwerk Management	von aussen
UDP, TCP	162	SNMP	Netzwerk Management	von aussen
UDP, TCP	445	NeTBIOS	Microsoft-DS, SMB	beide
TCP	524	tcpNCP	Netware NCP über TCP	beide
TCP	540	UUCP	Mail (zu Mailhosts durchlassen)	von aussen
TCP	1080	Socks Anwendungsproxy	von aussen	
TCP	1433	MS SQL	MS SQL Server	beide
TCP	1434	MS SQL	MS SQL Monitor	beide

Tabelle 13.1: Filterregeln des HRZ der TU Darmstadt

Beispiel für Filterregeln

Aus: Eckert, IT-Sicherheit

Firewalls

Purpose of a firewall:

- **A firewall controls and filters access to and from a protected network.**
  - Usually a firewall is placed between the trusted internal network (LAN, intranet) and the untrusted external network (Internet).
- **Securing a firewall:**
  - A firewall is a crucial component in the security perimeter. Thus a firewall should not run user programs such as web servers since these present a point of attack (attack vector).
- **Firewall types:**

  Over time different types of firewalls emerged.

  1. Packet filters (1st generation firewalls):
     ➔ Filter rules applied to individual packets
  2. Circuit level gateways (2nd generation firewalls):
     ➔ Monitor connections and flows of packets
  3. Application level gateways (3rd generation firewalls):
     Proxy, monitor and inspect the application traffic
  4. Stateful Multi-Layer Inspection Gateways SMLI:
     ➔ Combination of 1., 2. and 3.
  5. Distributed firewalls:
     ➔Host based firewall with central management
  6. Firewalls with NAPT:
     ➔Firewalls combined with NAPT functionality

[Quelle: Peter R. Egli, ZHAW Zürich]

Firewalls - Packet Filters

Function:

- **Simple packet filters** where the first firewall types that were deployed (1st generation firewall).

Packet filters usually run on layer 3 (IP) and execute the filter rules defined in ACLs (Access Control List). These firewalls are **stateless** because the filtering is applied to individual packets only (filtering does not depend on the state of previous packets of the same connection).

Filter rules:

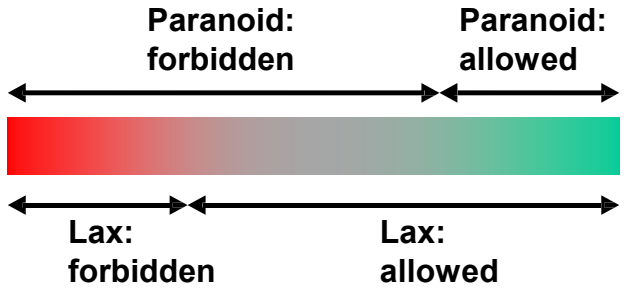
- The filter rules can range from simple source / destination IP address filtering to more complex rules that filter packets based on TCP flags, protocols or combinations thereof.
  - Filter on source / destination IP address
  - Filter on source / destination transport port
  - Filter on protocol such as ICMP, UDP, TCP, SCTP, IPSec
  - Filter on TCP flags (ACK, SYN, FIN, RST)
  - Filter on combinations of source/destination IP, source/destination port and protocol

Filter rule management:

- Over time the table with filter rules grows. Usually more rules are added than deleted.
- This may lead to large tables with rules that can not be deleted anymore because otherwise there may be the risk that a vital service is disabled.
- Therefore it is very important to document filter rules and tag them with a timestamp so it is possible in the future to decide whether a rule is still needed or not.

[Quelle: Peter R. Egli, ZHAW Zürich]

Filter policies:



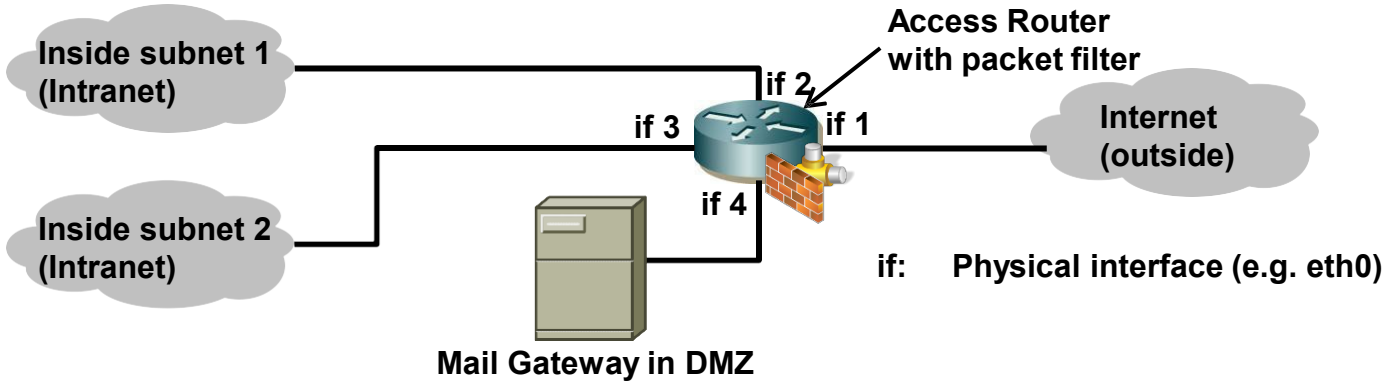
- **There are 2 main policies for filter rules:**
  - **Lax policy**: Allow everything that is not explicitly forbidden.
  - **Paranoid**: Forbid everything that is not explicitly allowed.
- The administrator has to trade-off security (paranoid policy) against usability (lax policy). If the firewall is too restrictive, there is the danger that employees establish ways around the firewall (rogue WLAN APs, SSH tunnels, HTTP tunnels).

Packet filters + NAPT:

- Packet filters are often combined with address and port translation (NAPT, NPAT).
  - NAPT is used to conserve public IP addresses.
  - Additionally NAPT provides a minimal level of security as it hides internal IP addresses.
- Where to use packet filters:
  - Packet filtering can be applied at many points (not only company firewalls). E.g. an ISP should install ingress filtering (perform filtering of traffic from customers into the Internet which has source addresses that do not belong to ISP, i.e. filter spoofed IP packets).

[Quelle: Peter R. Egli, ZHAW Zürich]

Filter rules consist of match elements (address, ports, flags etc.) and an action that defines what to do with a matching packet.



Example ingress filtering on if 1 (for ingress packets, i.e. outside→if1):

Action	Source IP	Source port	Dest. IP	Dest. port	Flags	Comment
block	*{subnet1}*	*	*	*		*Block spoofed packets.*
block	*{subnet2}*	*	*	*		*Block spoofed packets.*
block	*{subnet3}*	*	*	*		*Block spoofed packets.*
allow	*	*	*mail GW*	*25*		*Allow incoming SMTP to mail gateway.*
allow	*	*	*{subnet1}*	*	*ACK*	*Allow acks for outgoing TCP connections from subnet 1.*
allow	*	*	*{subnet2}*	*	*ACK*	*Allow acks for outgoing TCP connections from subnet 2.*

[Quelle: Peter R. Egli, ZHAW Zürich]

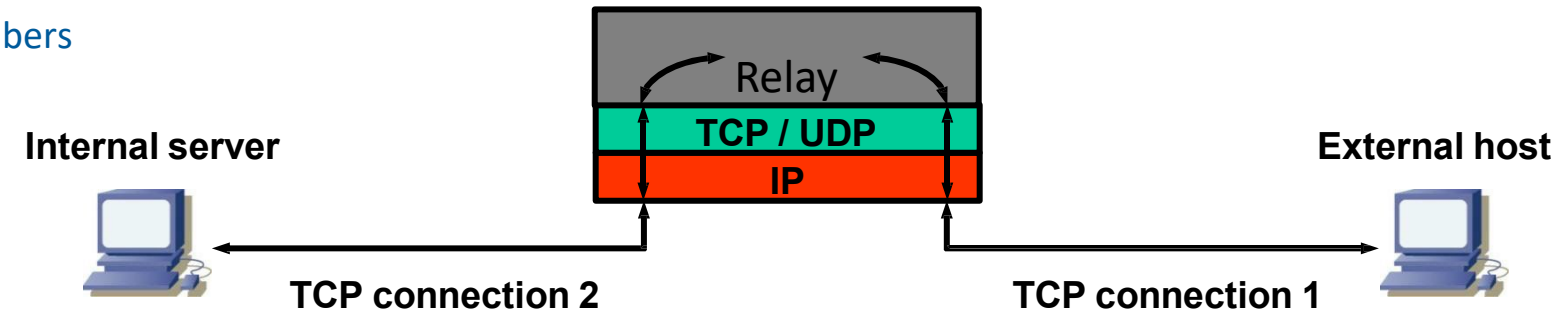
Firewalls - Circuit Level Gateways

Function:

- Circuit level gateways (2nd generation firewalls) relay incoming TCP connections and UDP sessions (similar to a proxy server).
- The gateway terminates (is endpoint of) the external TCP connection and establishes a new TCP connection to the internal host.
- Thus the internal host is not exposed and can therefore not be attacked,

i.e. sending malformed packets does not harm the internal host. IP tricks like wrong fragments and firewalking probes are terminated at the gateway and the internal host is protected.

- Additionally, a circuit level gateway closely monitors the external TCP connection (checks if it has a correct behavior):
  - TCP flags
  - Sequence numbers
  - ACK numbers



[Quelle: Peter R. Egli, ZHAW Zürich]

netfilter

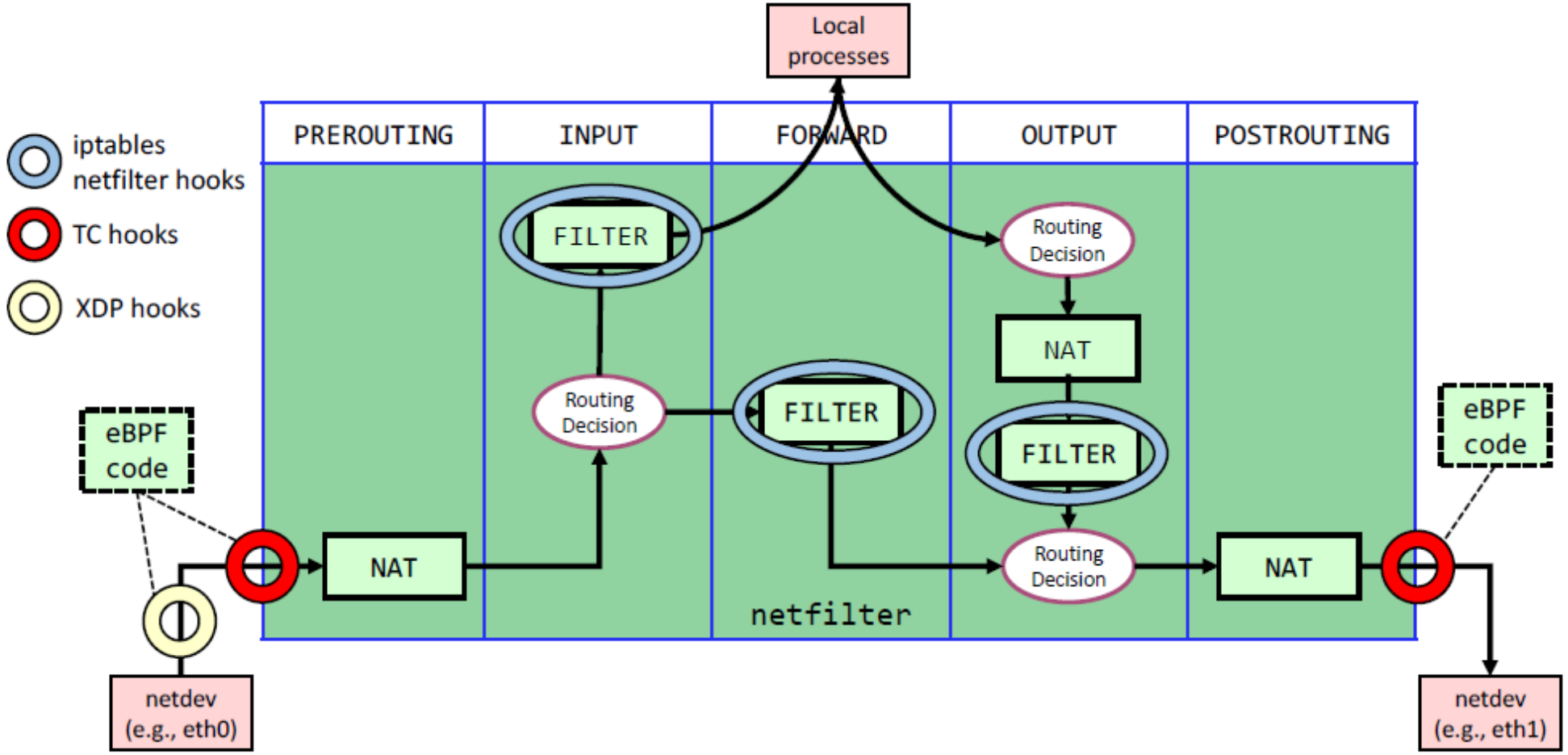


Figure 1: Comparing the location of netfilter and eBPF hooks.

Quelle: „Toward an eBPF-based clone of iptables“
Matteo Bertrone, Sebastiano Miano, Jianwen Pi, Fulvio Risso, Massimo Tumolo

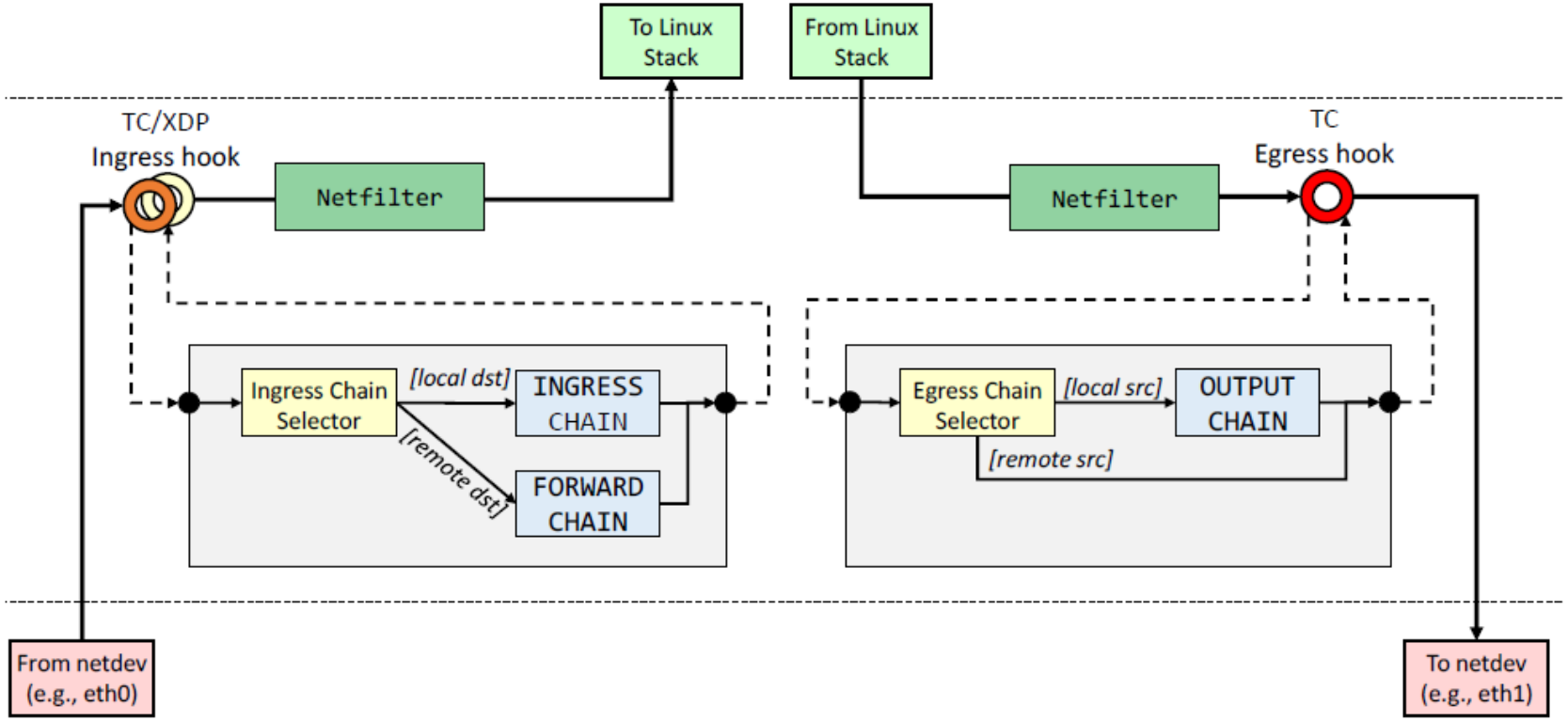


Figure 2: Main filtering architecture of bpf-iptables (without connection tracking).

Quelle: „Toward an eBPF-based clone of iptables“
Matteo Bertrone, Sebastiano Miano, Jianwen Pi, Fulvio Risso, Massimo Tumolo

Firewalls - Circuit Level Gateways

No routing allowed:

- **The circuit level gateway is not a router. Routing MUST be switched off.**
- **If routing is switched on, the packets bypass the relay and the internal host is unprotected.**
- **The gateway terminates TCP (and UDP) on one side and relays the payload to another TCP connection (or UDP packet) on the other side.**

- **Examples:**
  - SOCKS (RFC1928), tcprelay

[Quelle: Peter R. Egli, ZHAW Zürich]

Firewalls - Application Level Gateways

Function:

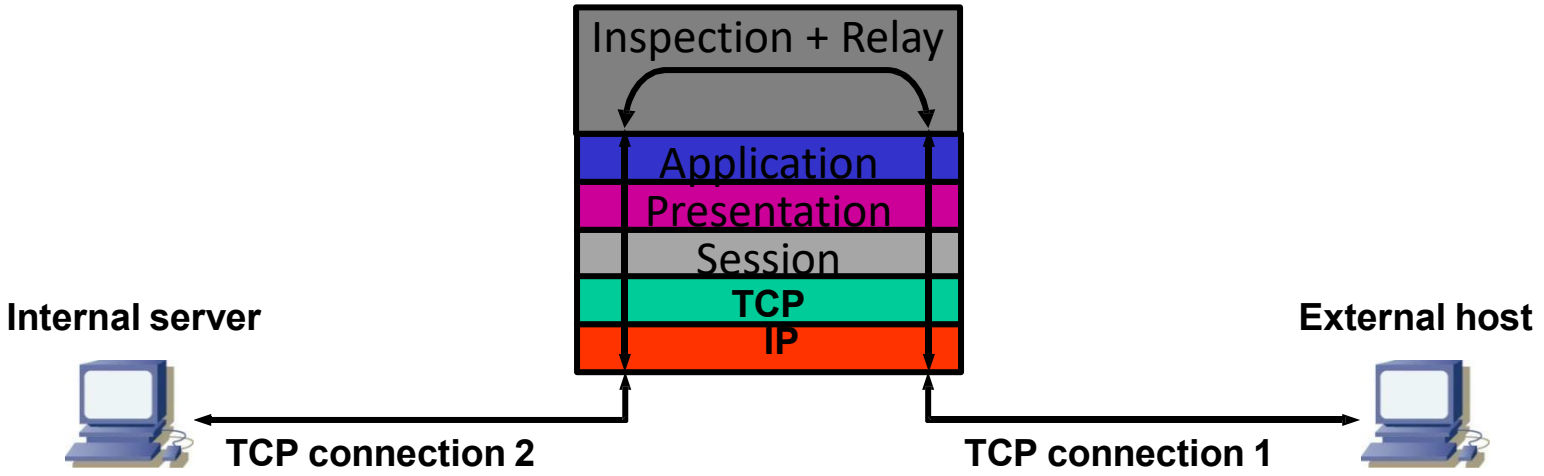
- Application level gateways (3rd generation firewalls) monitor the traffic up to the application layer. Thus application level gateways are proxies.
- Such gateways not only terminate transport connections, but also inspect the application traffic and only relay application payloads if these do not contain harmful data.
- **Examples of application level inspection functions:**
  - Filter dangerous content such as *.exe email attachments
  - Filter banned FTP commands like STOR, e.g. when company policy disallows uploads
  - Filter active web page content like Javascript
- **Application firewalls must 'understand' all application protocols that pass through the firewall. Therefore such firewalls are complex.**

Examples:

- Web Application Firewall (WAF),
- AppArmor (Linux kernel module)

[Quelle: Peter R. Egli, ZHAW Zürich]

Firewalls - Application Level Gateways



[Quelle: Peter R. Egli, ZHAW Zürich]

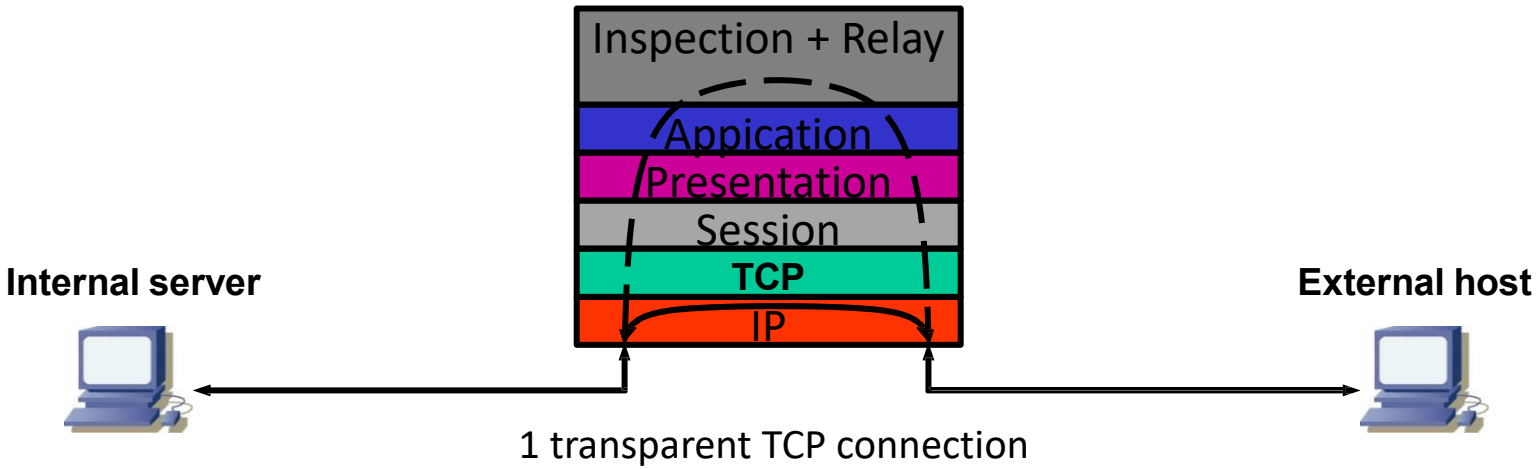
Firewalls – Stateful Multi-Layer Inspection Gateways (SMLI)

Function:

- SMLI combines packet filers, circuit level gateway and application level gateway. Such firewalls inspect the packets on OSI layer 2 through 7.
- All packets are checked if they belong to a transport connection or session (state table).
- SMLIs store the state of the connections and monitor it.
- UDP and ICMP packets are dynamically assigned to a connection (e.g. for a DNS query (UDP) the SMLI expects a corresponding DNS response packet (UDP).
- Additionally SMLIs perform application level filtering.
- Some protocols require the transport ports be dynamically opened. E.g. an FTP PORT or PASV command requires that the corresponding port be temporarily opened in the packet filter.

[Quelle: Peter R. Egli, ZHAW Zürich]

Firewalls – Stateful Multi-Layer Inspection Gateways (SMLI)



Firewalls – Distributed Firewalls

Function:

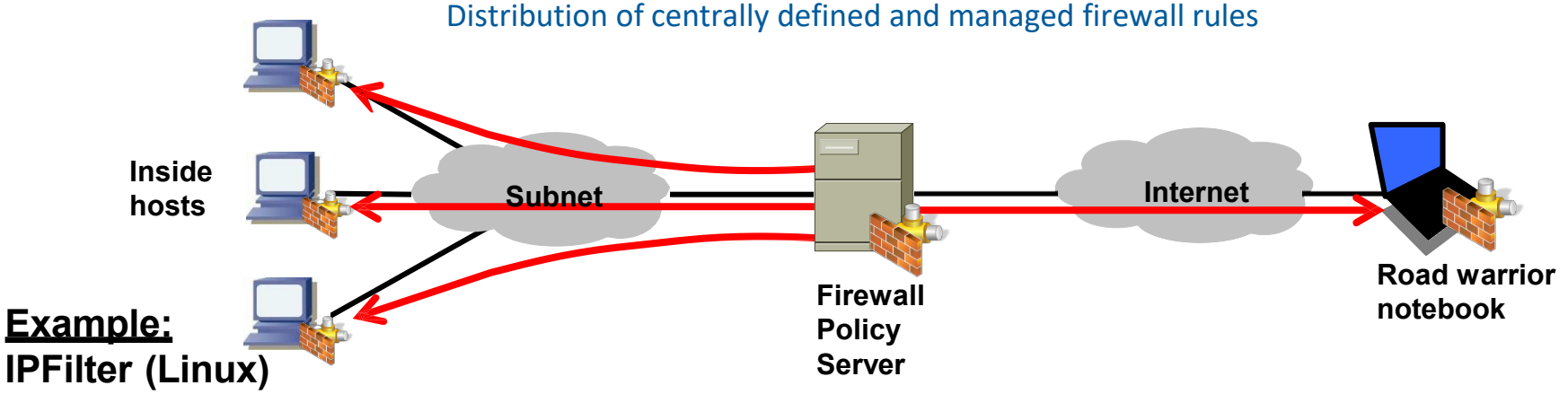
- Distributed firewalls are host-based firewalls, i.e. every host runs its own instance of firewall.

Distributed firewalls are similar to personal firewalls. The difference is that distributed firewalls are centrally managed, i.e. by centrally defining the filter rules and distributing them to all hosts.

Advantage:

- Standard firewall appliances have the drawback of being a single point of failure. If the firewall breaks the entire security may be compromised.
- Distributed firewalls protect each host individually.
- These firewalls allow to protect also hosts that or not inside a topologically isolated space (road warrior notebooks).

Distribution of centrally defined and managed firewall rules



Example:
IPFilter (Linux)

[Quelle: Peter R. Egli, ZHAW Zürich]

Firewalls with NAPT:

Simple NAT function (without port translation):

➔ **NAT for IP subnet address changes:**

- Simple NAT (Network Address Translation, i.e. replacement of an IP address by another one) may be used to change IP addresses in a subnet change without changing the routing entries in other subnets (IP address change is transparent to other subnets).

➔ **NAT for mapping public IPs to private IPs in a DMZ:**

- Another application of NAT is the placement of a firewall between public IPs (external network) and servers on a DMZ-network (e.g. 193.5.54.0/28 addresses (=public) are ‚NATed' to 10.0.0.0/28 addresses).

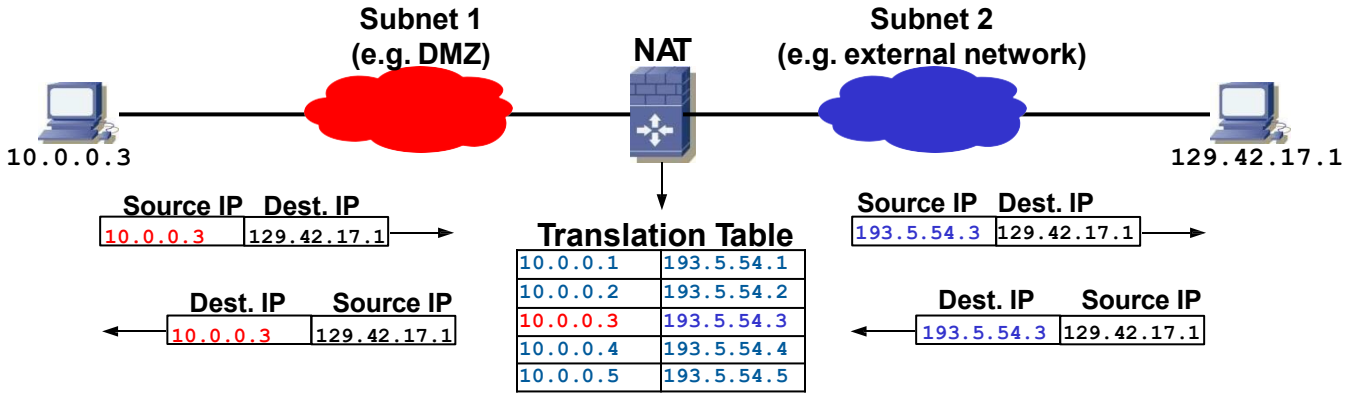
[Quelle: Peter R. Egli, ZHAW Zürich]

- **Static NAT:**

The translation table entries are set up statically (by the sysadmin). Each Subnet 1 host is assigned a specific entry in the translation table.

- **Dynamic NAT:**

Each time a packet from Subnet 1 arrives at the NAT, the latter chooses a free entry in the translation table. The mapping exists as long as there are packets for the host in Subnet 1 flowing back and forth. The entries are timestamped and aged out and ultimately deleted once they are no longer in use.

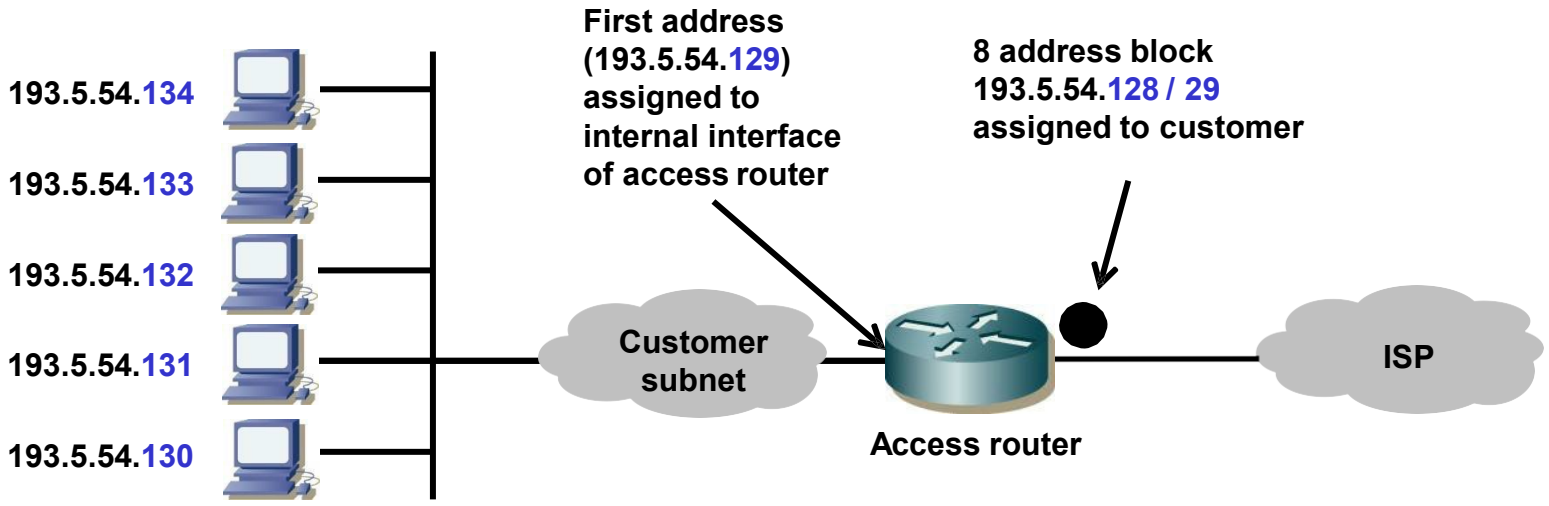


[Quelle: Peter R. Egli, ZHAW Zürich]

NAPT = Network Address and Port Translation:

Solution 1 for conserving IPv4 addresses:

- CIDR (Classless interdomain routing, RFC1517 et.al.) was introduced where IP addresses were no longer assigned as full class A, B or C networks but in smaller chunks, e.g. 10 IP addresses out of a class A range (thus classless).
- Internet routers needed to become able to route IP addresses from the same class to different locations (thus classless routing). This alleviated the problem but did not solve it.



[Quelle: Peter R. Egli, ZHAW Zürich]

NAPT = Network Address and Port Translation:

Solution 2 for conserving IPv4 addresses:

- The introduction of private IP addresses alleviated the problem further.

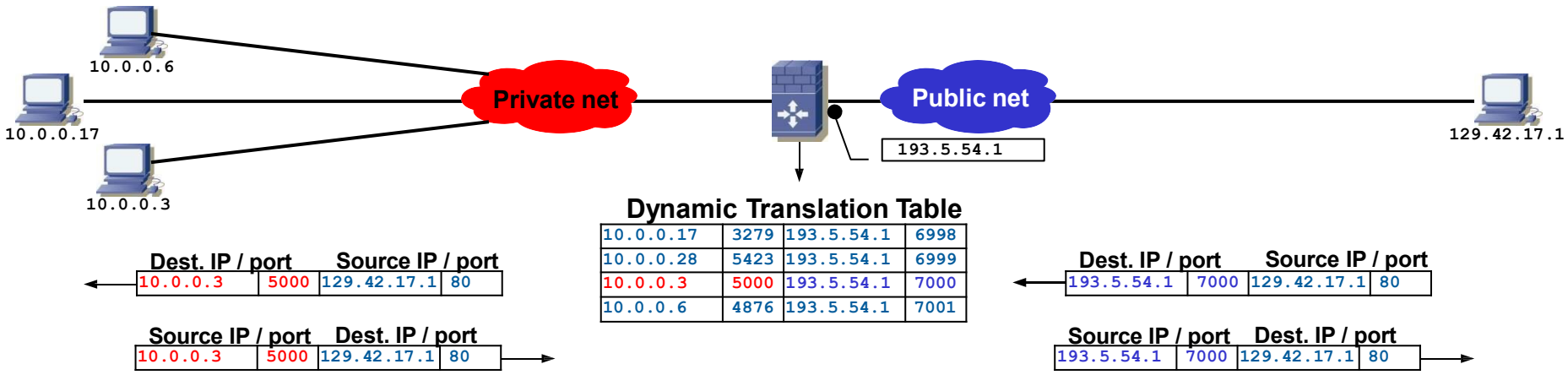
RFC1918 defines 3 ranges of IP addresses that were set aside and that can be used by every organization (no need to buy these addresses):

Range from	Range to	Mask	Hosts per network
10.0.0.0	10.255.255.255	255.0.0.0 (8bit)	16'777'216
172.16.0.0	172.31.255.255	255.240.0.0 (12bit)	1'048'576
192.168.0.0	192.168.255.255	255.255.0.0 (16bit)	65'536

- ➔ These addresses are not routed in the public Internet since they are private. Hosts with such IP addresses are either not reachable from the public Internet or need an address translation (NAPT)
- ➔ With NAPT it is possible to hide such hosts behind one single public IP address (which is on the NAPT box):

[Quelle: Peter R. Egli, ZHAW Zürich]

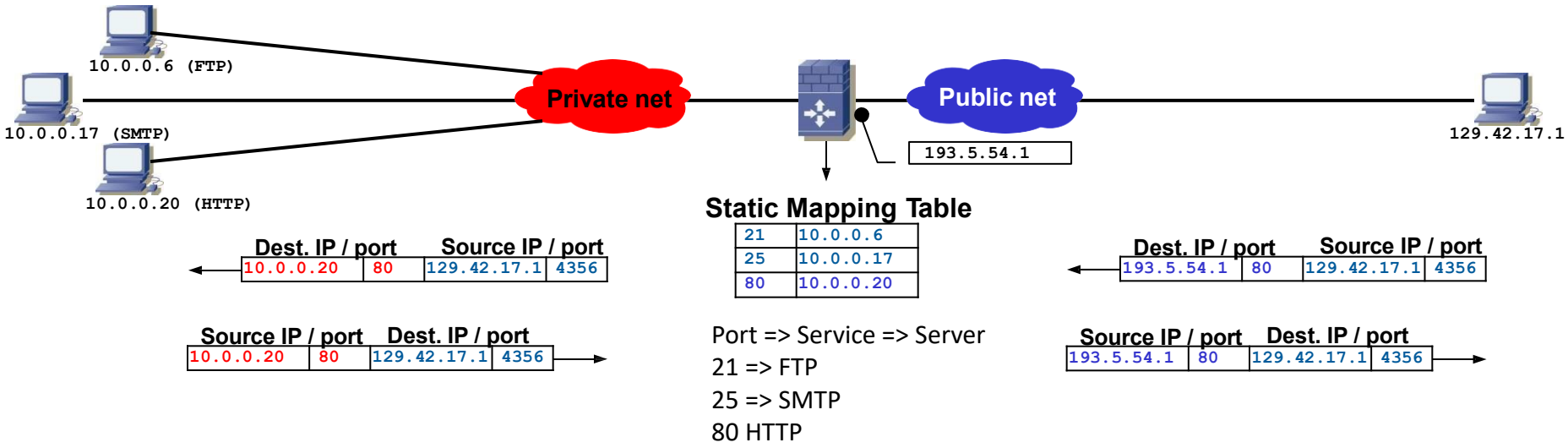
NAPT of private IP addresses



[Quelle: Peter R. Egli, ZHAW Zürich]

NAPT = Network Address and Port Translation:

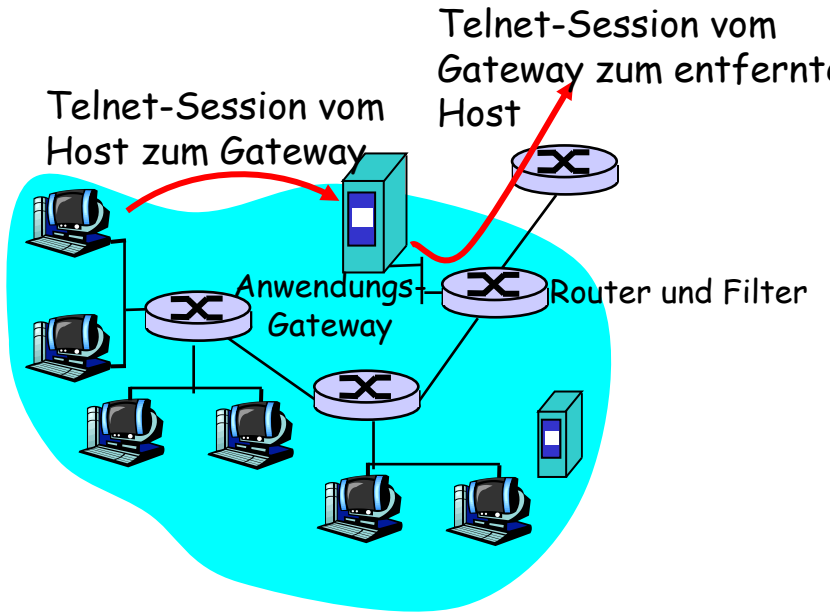
- **How to hide a server behind NAPT and still have the server accessible from the public Internet:**



[Quelle: Peter R. Egli, ZHAW Zürich]

Anwendungs-Gateways

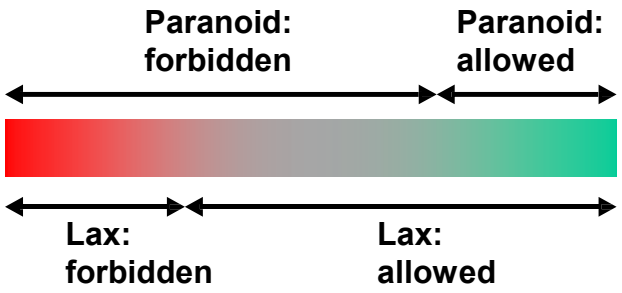
- **Pakete werden basierend auf Anwendungsdaten und aufgrund von IP/TCP/UDP-Feldern untersucht**
- **Beispiel: nur bestimmte interne Benutzer dürfen Telnet nach außen verwenden**



1. verlange, dass alle Telnet-Verbindungen über das Gateway laufen
2. für autorisierte Benutzer baut das Gateway eine Telnet-Verbindung zum Zielhost auf; Gateway vermittelt die Daten zwischen den beiden Verbindungen
3. Router-Firewall blockiert alle Telnet-Verbindungen, die nicht vom Gateway stammen.

Grenzen von Firewalls und Gateways

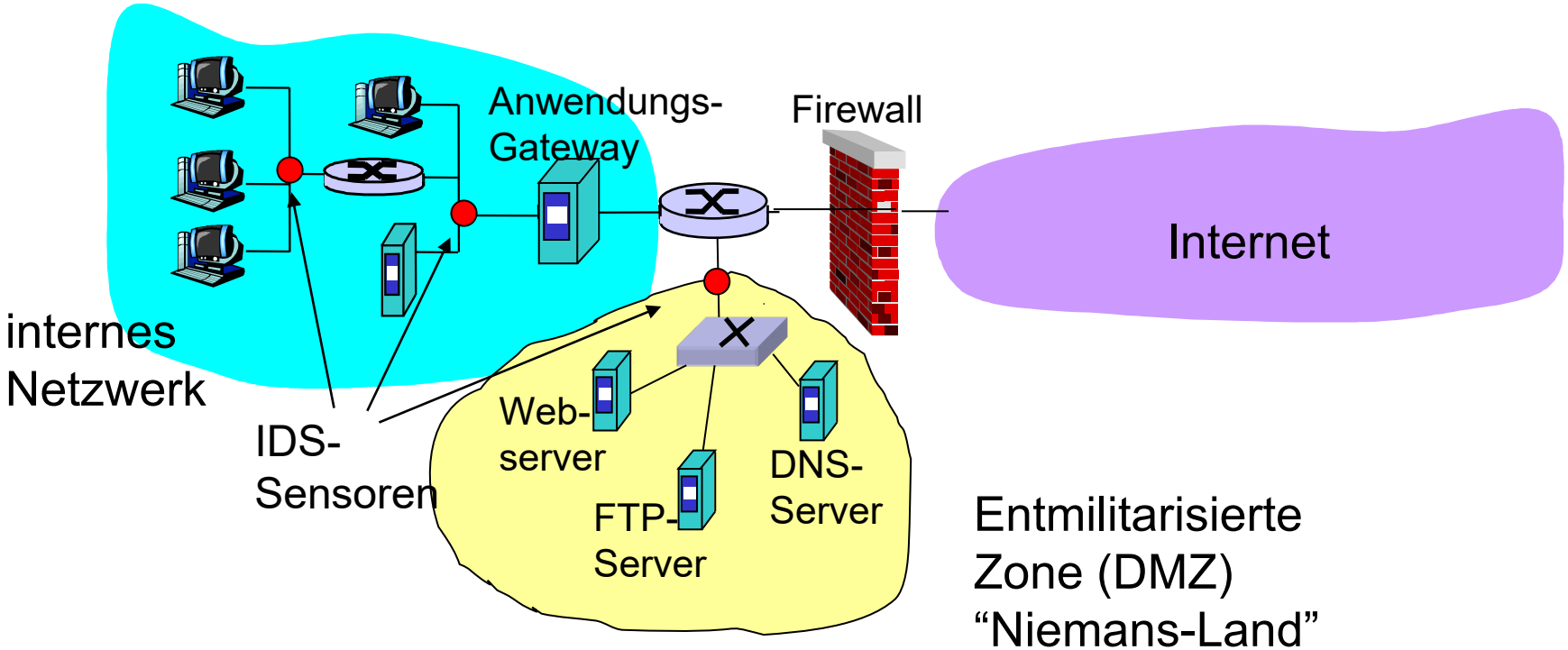
- **IP-Spoofing: ein Router kann nicht wissen, ob Daten “tatsächlich” von der angegebenen Quell-IP stammen**
- **wenn mehrere Anwendungen Spezialregeln brauchen, hat jede ein eigenes Anwendungs-Gateway**
- **Client-Software muss wissen, wie man das Gateway kontaktiert:**
  - z.B. muss im Browser die Adresse des Web-Proxy eingestellt werden
- **Filter verwenden oft eine “Alles-oder-Nichts”-Regel für UDP**
- **Abwägung: Kommunikationsmöglichkeiten mit der Außenwelt vs. Sicherheitslevel**
- **Policy:**
  - **Lax policy**: Erlaube alles, was nicht explizit verboten ist.
  - **Paranoid**: Verbiete alles, was nicht explizit erlaubt ist



Intrusion-Detection-Systeme

- **Paketfilter:**
  - arbeiten nur auf den TCP/IP-Headern
  - Daten unterschiedlicher Sitzungen können nicht korreliert werden
- **IDS: Intrusion-Detection-System**
  - **Deep Packet Inspection (DPI):** betrachtet den Paketinhalt (vergleicht z.B., ob im Paket Zeichenketten vorkommen, die in einer Datenbank bekannter Angriffe und Viren vorkommen)
  - Korrelation mehrerer Pakete erlaubt das Erkennen von
    - Port-Scanning
    - Netzwerk-Mapping
    - DoS-Angriffe

- **mehrere IDS: verschiedene Tests an verschiedenen Stellen im Netz**



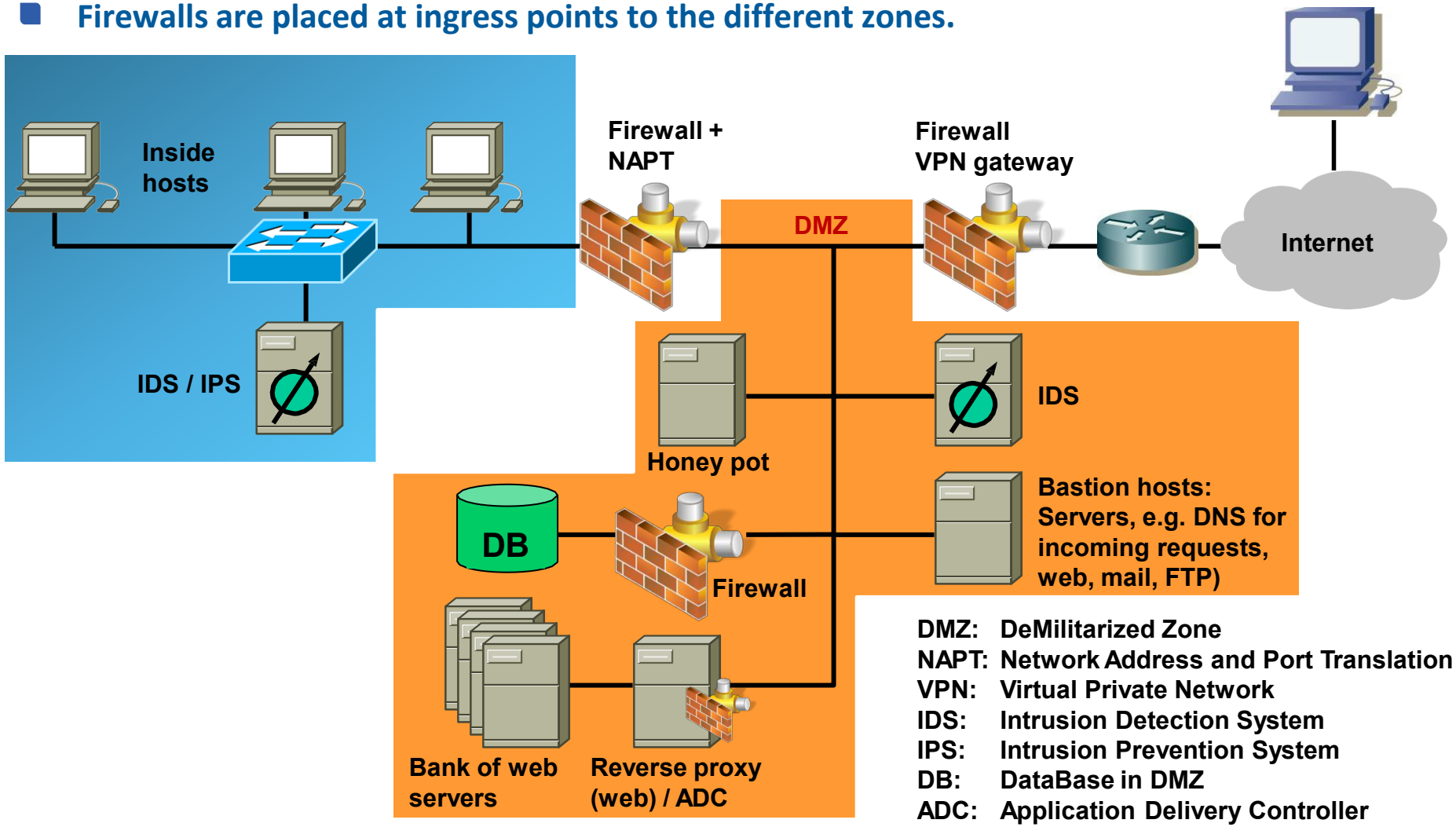
DeMilitarized Zone (DMZ)

- **Das Problem:**
- **Server mit Access vom Internet (web, mail, FTP) sind komplex und daher unsicher: Wohin?**
  - Ins Intranet hinter die Firewalls? Wenn diese Server gehackt werden, steht dem Hacker der Zugriff auf das Intranet offen.
  - Ohne Firewall direkt am Internet sich sie nicht geschützt.
- **➔ Eine spezielle Zone zwischen dem “wilden” Internet und den geschüzten Intranet:**
  - Isolierung der Systeme mit Zugang nach Außen
  - Anbindung an das offene Internet
  - Aber: Genau Überwachung des Verkehr
  - “Niemans-Land” => DeMilitarized Zone (DMZ)

- **Bereitstellung von Diensten (E-Mail, WWW, etc...) für WAN und LAN**
- **Server in der DMZ können von sich aus keine Verbindung zum LAN aufbauen**
- **Ein gehackter Server in der DMZ kann somit nicht das LAN kompromittieren**

Network Security Architecture

- The network is separated into an „inside“ (intranet) and „outside“ (DMZ and Internet).
- Firewalls are placed at ingress points to the different zones.



DMZ: DeMilitarized Zone
NAPT: Network Address and Port Translation
VPN: Virtual Private Network
IDS: Intrusion Detection System
IPS: Intrusion Prevention System
DB: DataBase in DMZ
ADC: Application Delivery Controller

[Quelle: Peter R. Egli, ZHAW Zürich]

Components of network security architecture:

- **Firewalls:**
  - Firewalls are placed at ingress / egress points of networks or subnets.
  - Firewalls may be combined with the NAPT function. The use of private IPv4 addresses (IPv6: unique local addresses) avoids that packets leak into the Internet due to wrong routes.
- **DMZ (DeMilitarized Zone):**

  Problem / dilemma:
  - Servers with access from the Internet (web, mail, FTP) are complex and thus inherently insecure (provide more attack vectors).
  - If these servers are placed inside (Intranet), they are best protected (firewall) but if the servers are compromised (hacked), the hacker has access to the intranet.
  - If they are placed outside (direct access from Internet without firewall) they are least protected.
- **➔ Thus a special zone („no man‘s land“ between „wild“ public Internet and protected intranet) is created which is carefully monitored as it contains the servers that are exposed to the outside.**

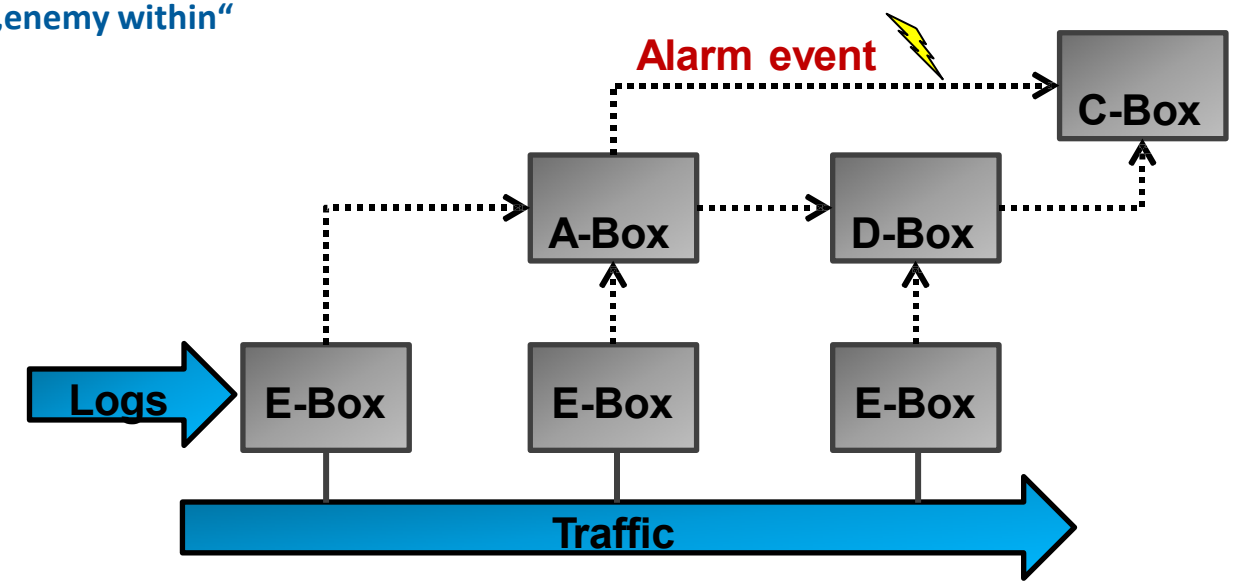
[Quelle: Peter R. Egli, ZHAW Zürich]

IDS - Intrusion Detection System (1/2):

- **An IDS complements a firewall by monitoring the network for (suspicious) traffic (e.g. firewalking, port scans, spoofing, unusually small TTL values, IP fragments, overlapping TCP segments, network sniffers etc.).**
- **Conceptually an IDS consists of an A-, C-, D- and E-box each with a different function.**
  - A-box: Network activity **A**nalysis (on data gathered by E-box and stored in D-box)
  - C-box: **C**ountermeasure mechanism (IPS, modify filter lists in routers etc.)
  - D-box: Storage mechanisms (loggers, store **D**ata produced by E- and A-boxes)
  - E-box: **E**vent generators (sensors)
- **IDS also protect against „enemy within“**

(it is estimated that 80%

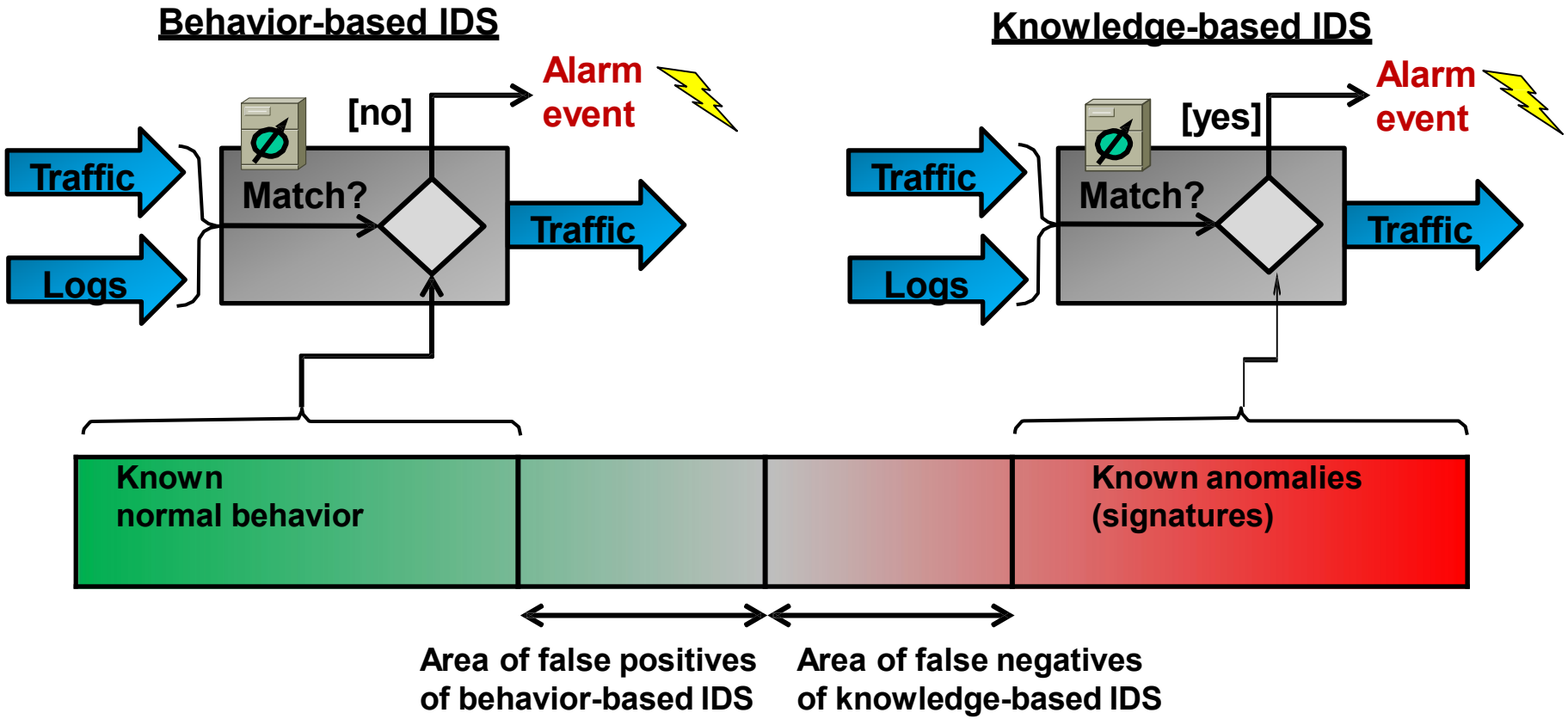
of attacks come from within).



[Quelle: Peter R. Egli, ZHAW Zürich]

IDS - Intrusion Detection System (2/2):

- With regard to functionality, IDS can be classified into:
  - Behavior-based IDS (=anomaly-based IDS)
  - Knowledge-based IDS (=signature-based IDS

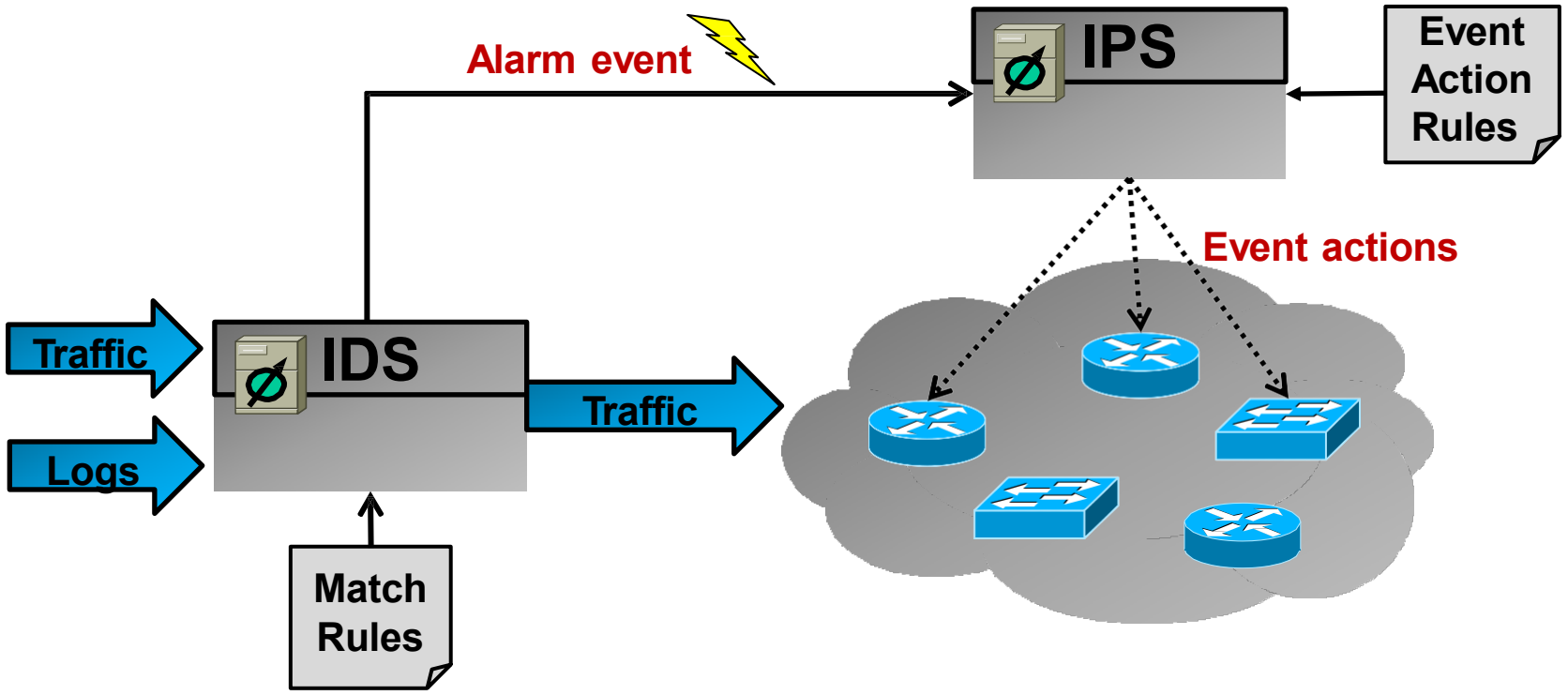


[Quelle: Peter R. Egli, ZHAW Zürich]

IPS - Intrusion Prevention System:

- An IPS complements an IDS (or is part of an IDS) by actively taking actions upon a detected intrusion.

Example actions are changing firewall filter rules, blocking traffic and resetting TCP Connections



[Quelle: Peter R. Egli, ZHAW Zürich]

Network Security Architecture

Honey pot:

- A honey pot is a decoy or booby trap to tie down an attacker and gather information about the attacker (example honey pot SW: honeyd, snort).
- Honey pots must be used with care and need to be well secured so they do not become security risk (present another attack vector).

Reverse proxy:

- A reverse proxy is used to hide web servers from the public Internet. Only the reverse proxy is visible to the outside and relays the requests to internal server(s).
- Reverse proxies control the access to web servers (e.g. block web mail access), but are also used for **caching** and **load balancing**.
- An **ADC** (Application Delivery Controller) is a special network appliance that acts as a reverse proxy but fulfills additional functions like load balancing, web application filtering (XSS, SQL injection etc.), security through protocol termination (HTTP, TCP, FTP etc.), compression, caching, failover, SSL offloading (termination of SSL connections) and connection pooling (save connection resources).

Bastion host:

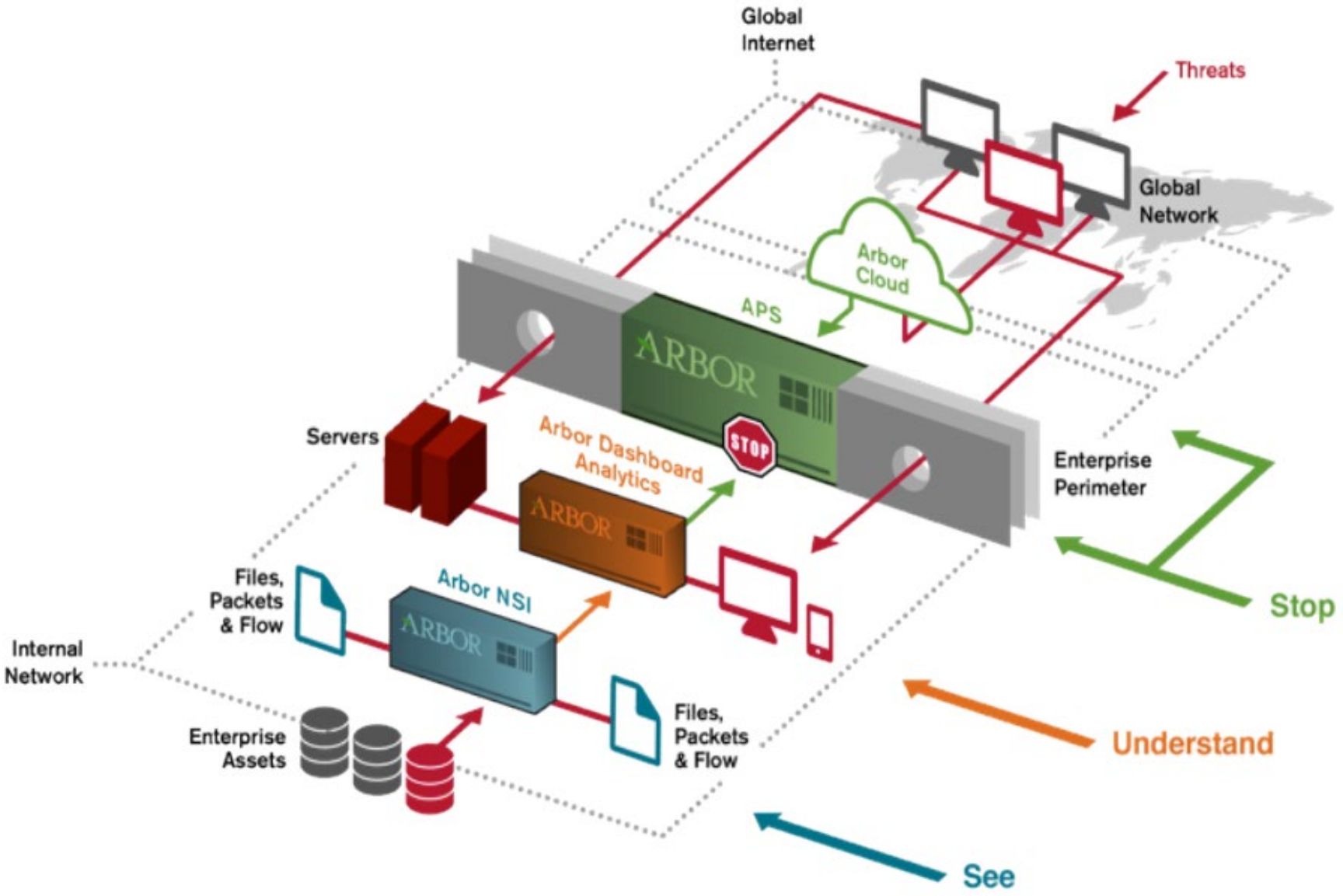
- Bastion hosts are servers that are exposed to the public Internet, i.e. provide services like DNS, web, mail or FTP to the Internet. Usually they are placed into the DMZ.
- Bastion host servers need to be specially secured (hardened servers).

No backdoors!

- Avoid Intranet Access via WLAN, PBX-Modems, bypassing the security walls!!

[Quelle: Peter R. Egli, ZHAW Zürich]

Beispiele zu Sicherheitslösungen: Arbor

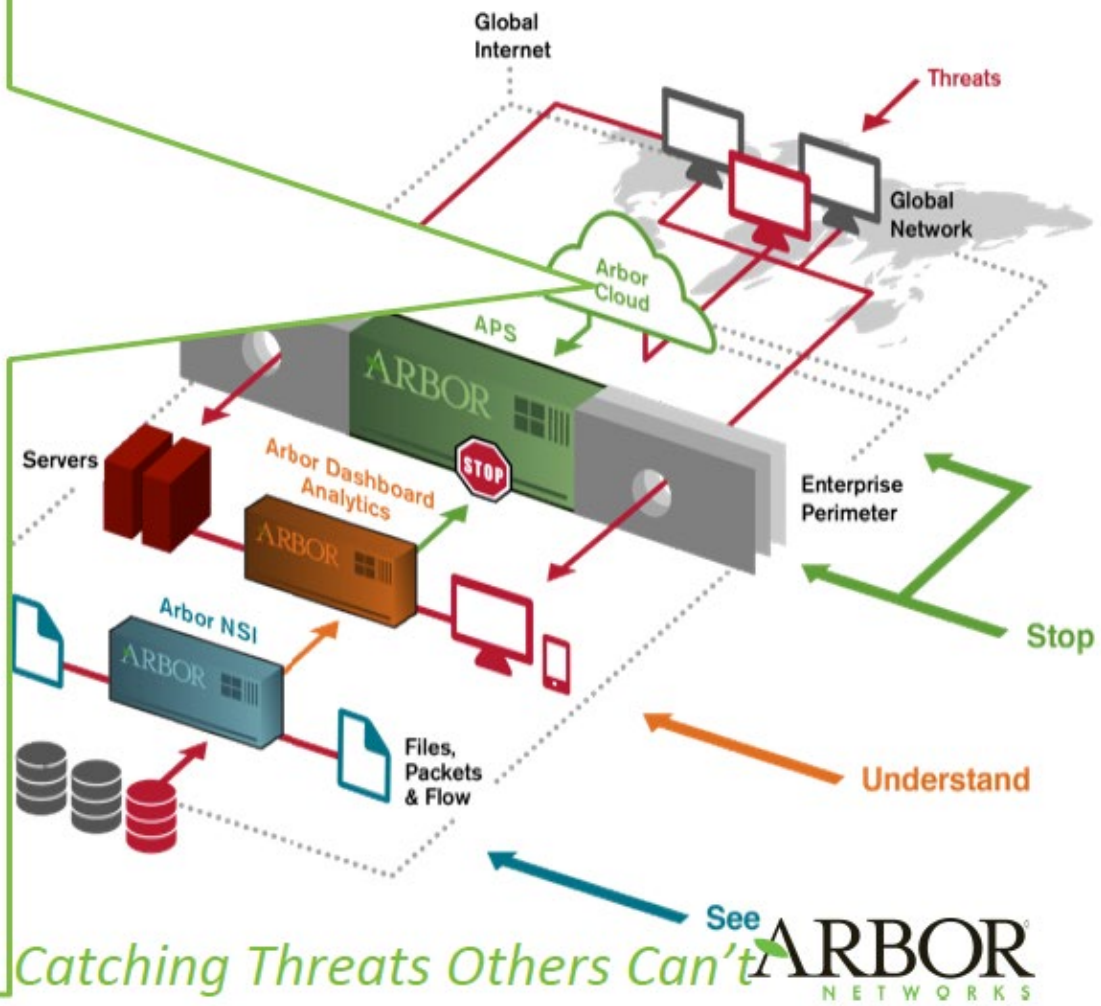


[Quelle: Arbor Networks General Overview]

Beispiele zu Sicherheitslösungen: Arbor

Arbor Cloud

- Arbor Supported (Arbor SOC)
- Integrates with Pravail APS
  - Accepts Cloud-Signaling
- Volumetric & Application attacks
- Pricing based on volume of peace-time (clean) traffic
- Global Cloud Scrubbing Capacity
  - 4 Global Scrubbing Centers
  - Dual Tier-1 ISPs
  - 100% Arbor mitigation equipment
  - 1 TERRA Capacity (Scalable)
- BGP and/or DNS Diversion Options
- SSL decryption option
  - Only with DNS Service

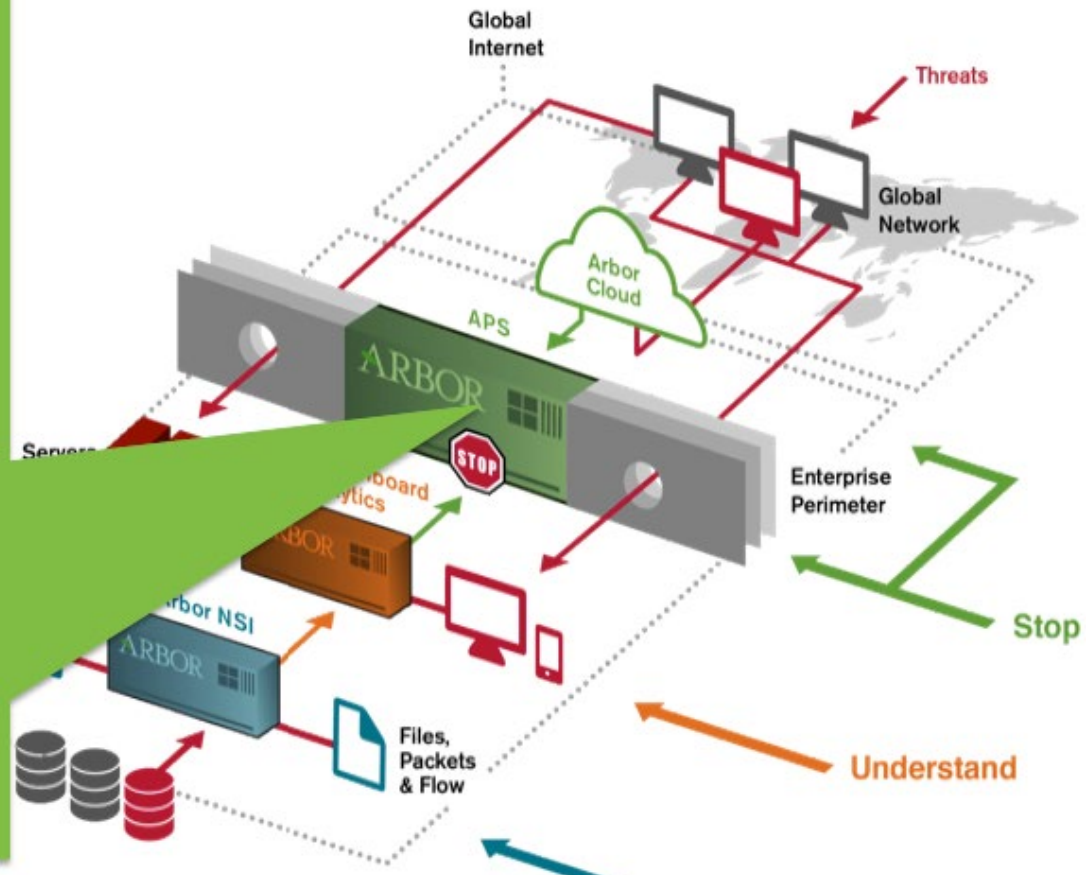


[Quelle: Arbor Networks General Overview]

Beispiele zu Sicherheitslösungen: Arbor

Pravail Availability Protection System (APS)

- Immediate protection from current threats.
  - Utilise ATLAS threat intelligence to protect your organisation from the latest threats.
- Easy to install and deploy
  - Easy to operationalize and deploy. Built in bypass functionality. Detailed traffic and reporting for advanced users.
- (Arbor) Cloud Signaling
  - Integration with cloud based DDoS protection services to provide the automated, layered protection necessary to deal with multi-vector attacks.



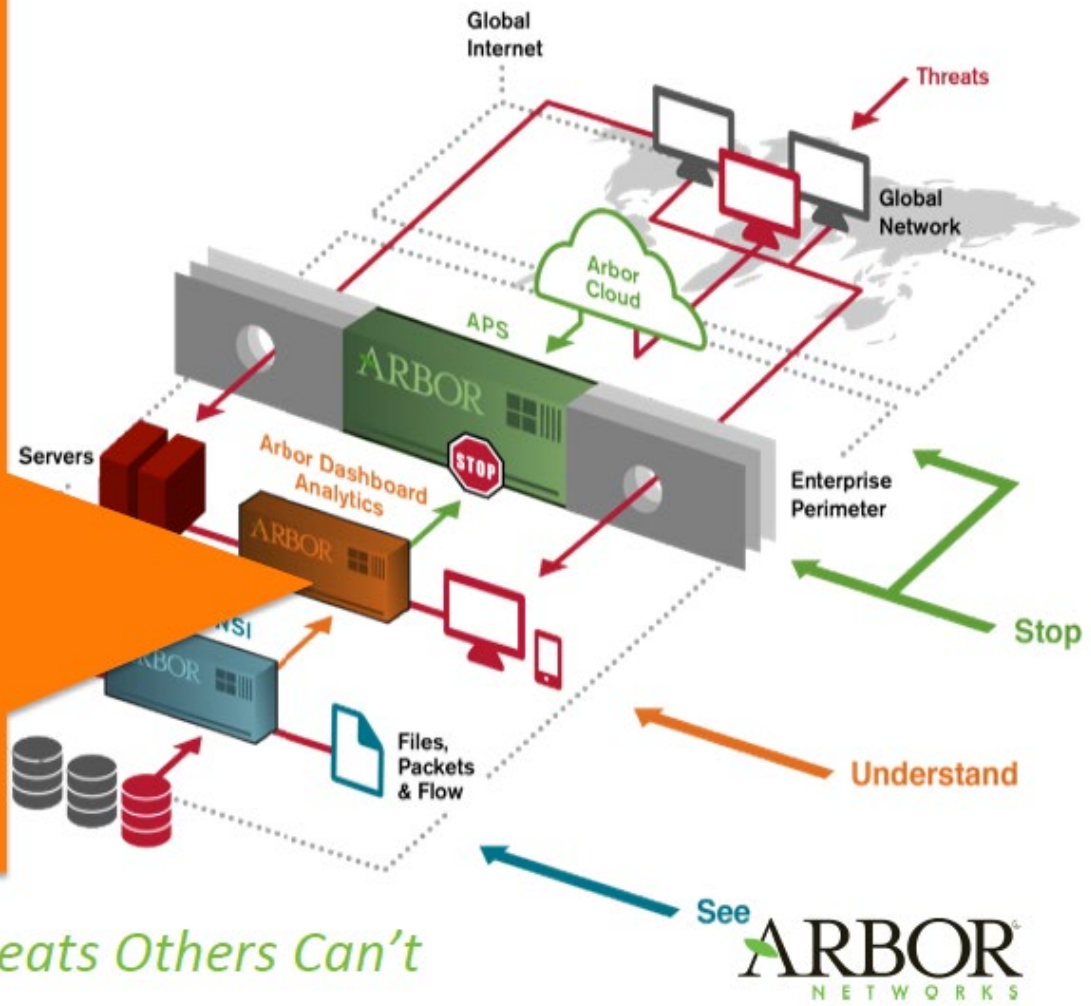
Catching Threats Others Can't

[Quelle: Arbor Networks General Overview]

Beispiele zu Sicherheitslösungen: Arbor

Pravail Security Analytics

- **Unique full-packet capture and security analytics solution**
  - **Pre-configured to be setup quickly, easy to manage and intuitive to use**
- **Capability to analyze years / terabytes of network information flow data**
  - High Definition view of traffic, user can interact with data
  - Capture all external or internal traffic—capture points can be created anywhere within the network
  - Re-apply current threat intelligence data to historical traffic.

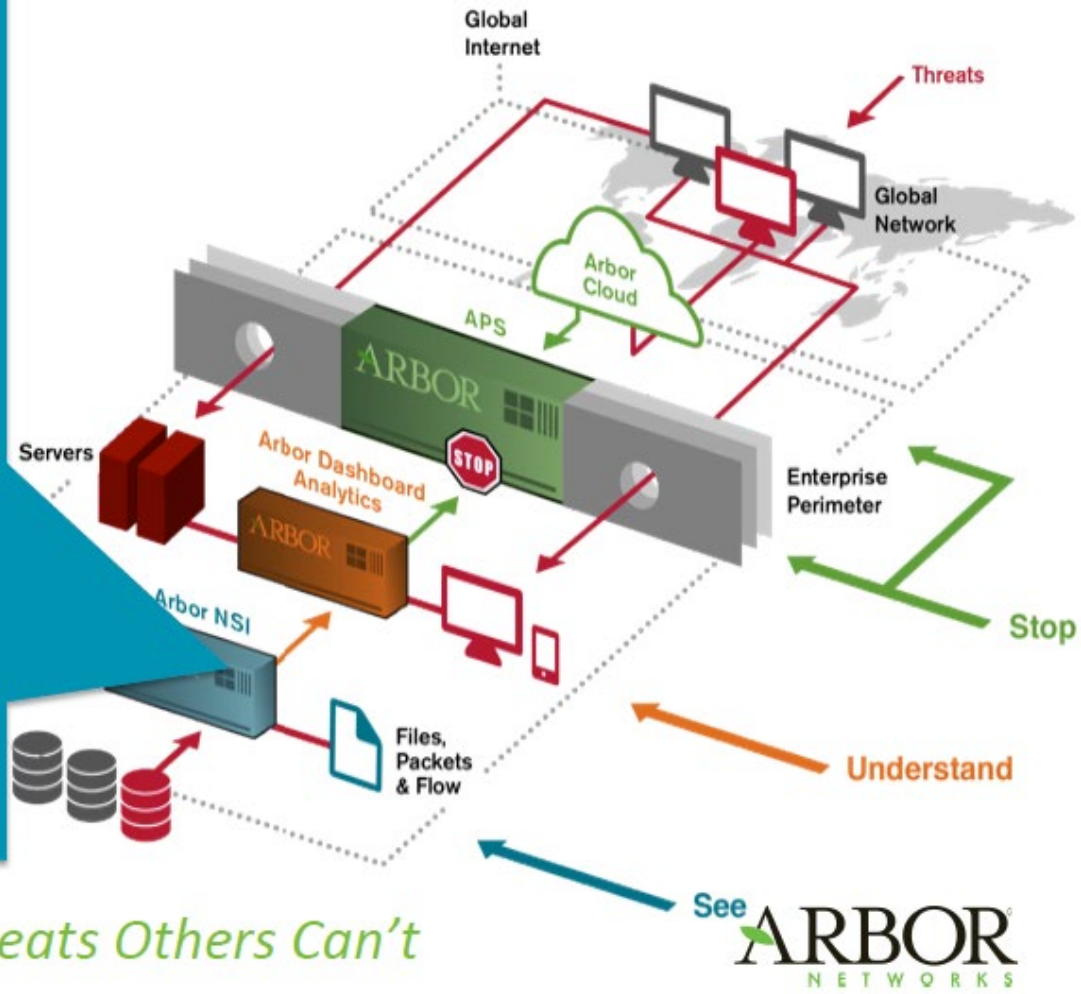


[Quelle: Arbor Networks General Overview]

Beispiele zu Sicherheitslösungen: Arbor

Pravail Network Security Intelligence (NSI)

- **Advanced Threat Detection**
  - Utilise ATLAS threat intelligence to protect your organisation. Profile critical systems and identify suspicious or unusual behavior wherever it occurs.
- **Enterprise-Wide Visibility**
  - Know your network; see who talks to who, when and how much. Identify tunneled, encrypted traffic leaving your network. Assign user-identities to network transactions.
- **Context & Forensics**
  - Understand the context for any detected event. Detailed log of historical network traffic.



Catching Threats Others Can't

[Quelle: Arbor Networks General Overview]

Some security guidelines (1/2)

1. **Use DMZ:**

   Place servers (HTTP, SMTP, FTP) into a DMZ.

2. **Block incoming HTTP traffic:**

   Block incoming HTTP and HTTPs (into LAN) unless really needed. This will foil attempts to tunnel protocols through the firewall through tunnels, e.g. using SOAP (port 80) or SSL (port 443). SOAP (Simple Object Access Protocol) = „anything over HTTP“!

3. **Restrict inbound HTTP traffic to web server:**

   Only the web server should be allowed to receive inbound HTTP.

4. **Minimize services:**

   Switch off any services / servers that are not needed. Every service / server represents a potential security hole (attack vector).

5. **Security by obscurity:**

   Do not disclose information if not necessary. Hide as much information (server names, addresses, locations, operating systems, user names etc.) as much as possible, but do not rely on it. It is just another layer in the security perimeter.

6. **Least priviledge:**

   Give services / servers and users least priviledge. If not necessary do not run servers / services as root. Otherwise if such a service / server is compromised the attacker gains root access.

[Quelle: Peter R. Egli, ZHAW Zürich]

Some security guidelines (2/2)

7. **Log service:**

   Run logging service to log unusual activities in order to identify potential attacks. Use a separate disk partition for the log files in order to protect against DoS attacks where the log files swamp the entire machine.

8. **Security vs. usability tradeoff:**

   Find a good tradeoff between security and usability. If users are disgruntled because they are unable to use certain services (e.g. unable to use FTP for downloading) they will find their way around the firewall and protection mechanisms (e.g. carry files on virus-infested storage devices into the company).

9. **Don't be an attacker, don't be a victim:**

   Avoid becoming a victim, but also an attacker (switch off services / servers that could become tools for the attacker for a dDoS attack). E.g. configure your mail server such that it can not become an open relay.

10. **Stay informed:**

    Stay informed about security problems. Consult web resources such as CERT Computer Emergency Response Team http://www.cert.org/ frequently.

[Quelle: Peter R. Egli, ZHAW Zürich]