Kapitel 2: Netzwerksegmentierung mit Firewalls und IDS-Systemen

Lernziele:

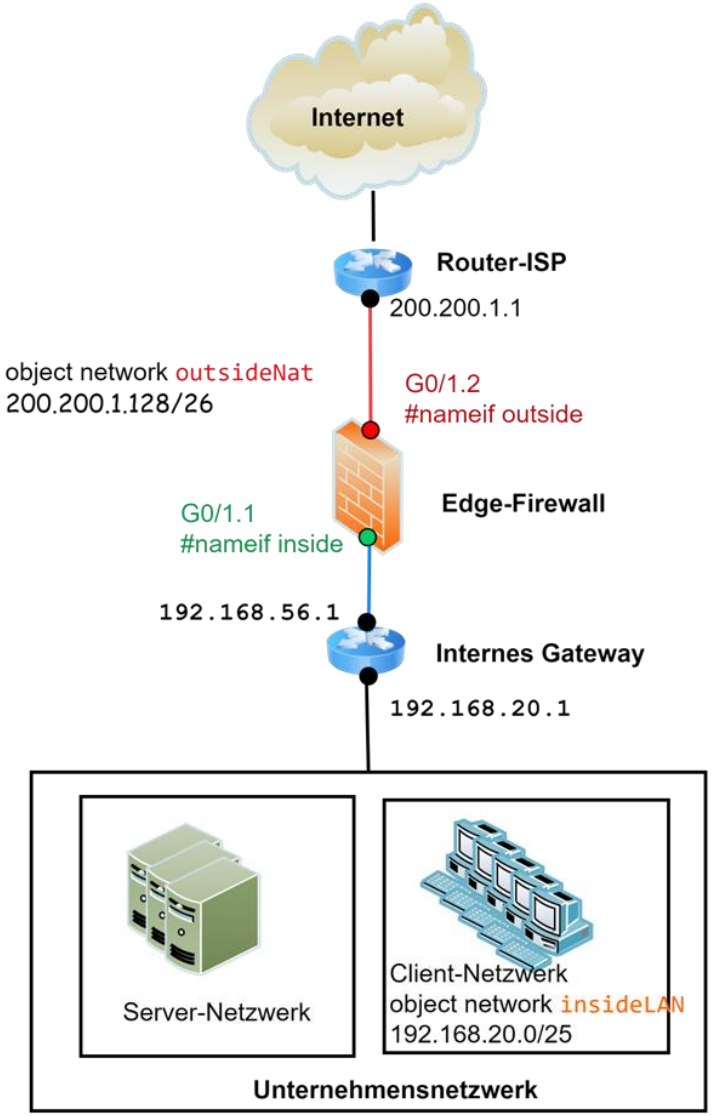
- Prinzip einer Firewall erklären und konfigurieren können.
- Unterschiede zwischen Packet- und Stateful-Firewall an Beispielen erklären können.
- Applikations-Firewalls erklären und konfigurieren können.
- Bedeutung eines SIEM-Systems
- IDS/IPS-Systemen erklären und konfigurieren können.

Überblick:

2.1 Firewalls

Ziele, Konfiguration, Ausprägungen,

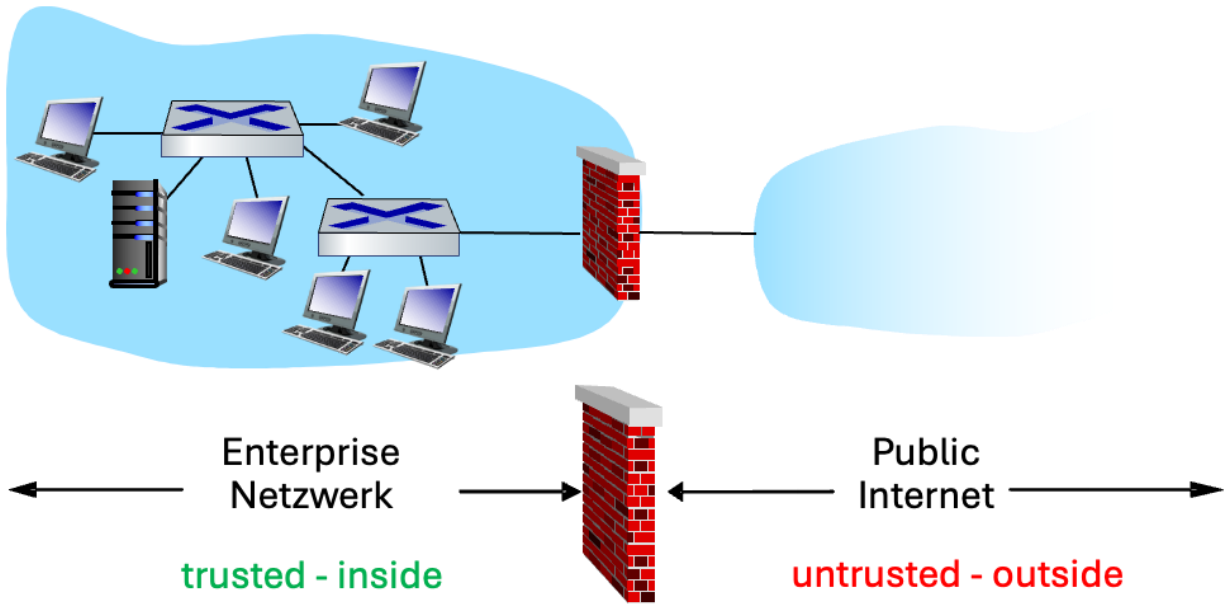
Platzierung, Beispiele



Firewall – Grundkonzept einer Edge-Firewall

Eine Firewall ist ein Netzwerkgerät, das den Zugriff auf ein Netzwerk vermittelt und bestimmte Zugriffsarten auf der Grundlage einer konfigurierten Sicherheitsrichtlinie erlaubt oder verbietet.

- Grundidee: Schutz eines vertrauenswürdigen Netzwerkes vor einem nicht vertrauenswürdigen Netzwerk durch ein definiertes Regelwerk.
- Inline: Der gesamte Datenverkehr wird von innen (Unternehmensnetzwerk) nach außen (Internet) und umgekehrt durch die Firewall geleitet werden.
- Nur autorisierter Datenverkehr, wie in der lokalen Sicherheitsrichtlinie für die Firewall definiert, wird weitergeleitet.
- Firewalls, die sich am Übergang vom Unternehmensnetzwerk in das Internet befinden, werden auch als Edge-Firewalls bezeichnet.

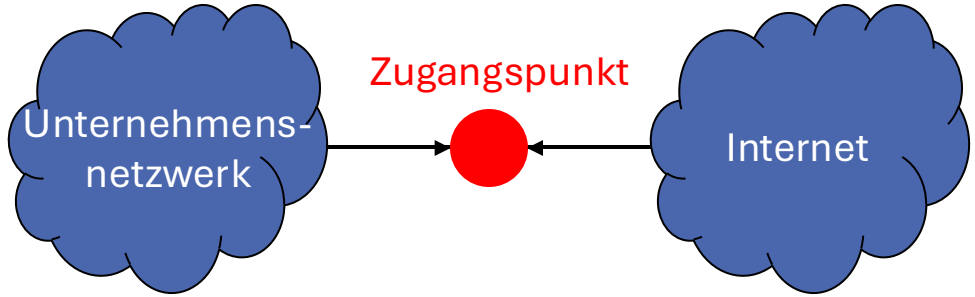


- Die Firewall selbst ist gehärtet, so das Sie resistent gegen Angriffe oder unerlaubte Eindringen von außen ist:
  - gehärtetes Betriebssystem und gehärtete Software (Secure SDLC, nur benötigte Dienste).
  - gehärtete Hardware (nur benötigte Schnittstellen, Schutz gegen Manipulation mittels Sensoren)

Edge-Firewall-Möglichkeiten

- Eine Edge-Firewall definiert einen einzelnen Zugangs-Punkt, durch den ein Netzwerkpaket vermittelt werden muss, um Dienste im Internet oder im Unternehmensnetzwerk zu erreichen.
- Die Verwendung eines einzelnen Zugangspunkts führt zu einem zuverlässigen und wirksamen Schutz:
  - Die Sicherheitsfunktionen werden auf einem einzelnen System eingerichtet, gepflegt und getestet werden.
  - Die Überwachung von sicherheitsrelevanten Ereignissen im eingehenden Datenverkehr erfolgt an einem zentralen Punkt.
  - Logs und Alarme können vom Firewall-System an ein zentrales SIEM-System weitergeleitet werden.

- Die Firewall kann verschiedene Flussrichtungen der Netzwerkpakete kontrollieren
  - Von innen nach außen blockiert die FW den Datenverkehr einer Malware, die eine Verbindung zu einem C2C-Server aufbauen möchte
  - Von außen nach innen blockiert due FW den unautorisierten Zugriff auf Dienste im Unternehmensnetzwerk.

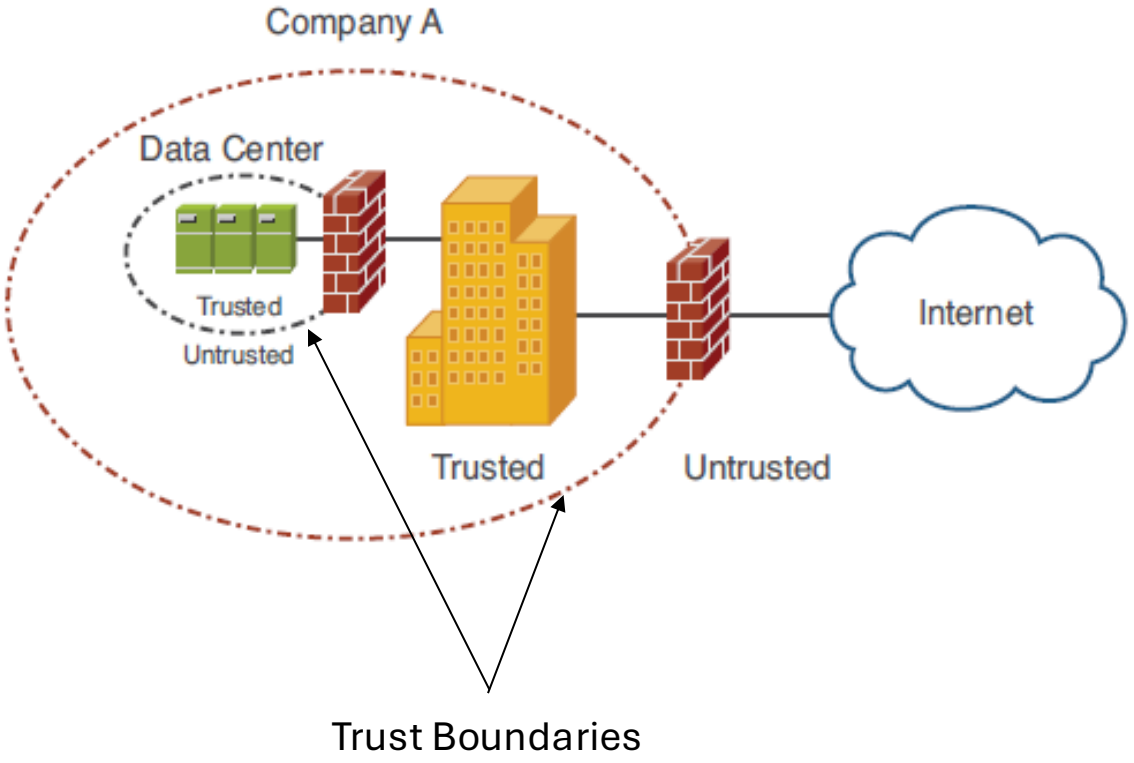


Grenzen einer Edge-Firewall

- Auch Firewalls haben ihre Grenzen, darunter die folgenden:
  - Besitzt eine FW eine SW-Schwachstelle, kann die Firewall möglicherweise umgangen oder sogar in die Kontrolle eines Angreifers gebracht werden.
  - DDoS-Angriffe durch Botnetze führen zu einer Überlastung der FW oder zur Überlastung der Internetleitung.
  - Die Firewall kann nicht vor Angriffen schützen, die die Firewall umgehen. Interne Systeme verfügen möglicherweise über Dial-Out-Funktionen, um eine Verbindung zu einem ISP herzustellen.
  - Die Firewall schützt nicht vor internen Bedrohungen, wie z. B. einem verärgerten Mitarbeiter oder einem Mitarbeiter, der unwissentlich mit einem externen Angreifer zusammenarbeitet.
  - Auf ein nicht ausreichend gesichertes WLAN kann von außerhalb der Organisation zugegriffen werden. Eine interne Firewall, die Teile eines Unternehmensnetzwerks trennt, kann nicht vor drahtloser Kommunikation zwischen lokalen Systemen auf verschiedenen Seiten der internen Firewall schützen.
  - Ein Laptop, PDA oder tragbares Speichergerät kann außerhalb des Unternehmensnetzwerks verwendet und infiziert und dann intern angeschlossen und verwendet werden.
  - Ein Hacker gelangt auf den Campus einer Firma und verbindet seinen Laptop unerlaubt mit dem Campus-Netzwerk.
  - Versteckte Malware durch Kodierung (URL-Encoding, BASE64-Encoding, ...) , Verschlüsselung oder durch versteckte Anweisungen im Payload der Nachrichten.

Firewall – Interfaces: Innen und Außen

- Eine Firewall besitzt grundsätzlich ein externes Interface („out-side“) und ein internes Interface („inside“).
  - Das externe Interface ist mit dem weniger vertrauenswürdigen Netzwerk verbunden.
  - Die interne Schnittstelle ist mit dem vertrauenswürdigen und zu schützenden Netzwerk verbunden.
- Auch innerhalb eines Unternehmensnetzwerkes kann es vertrauenswürdige und weniger vertrauenswürdige Bereiche geben, sodass auch an internen Vertrauensgrenzen ("Trust Boundaries") Firewalls zum Einsatz kommen.
- Beispiel:
  - Übergang Client-Netzwerk zu DataCenter-Netzwerk. DataCenter ist vertrauenswürdiger als das Netzwerk für die Anwender.

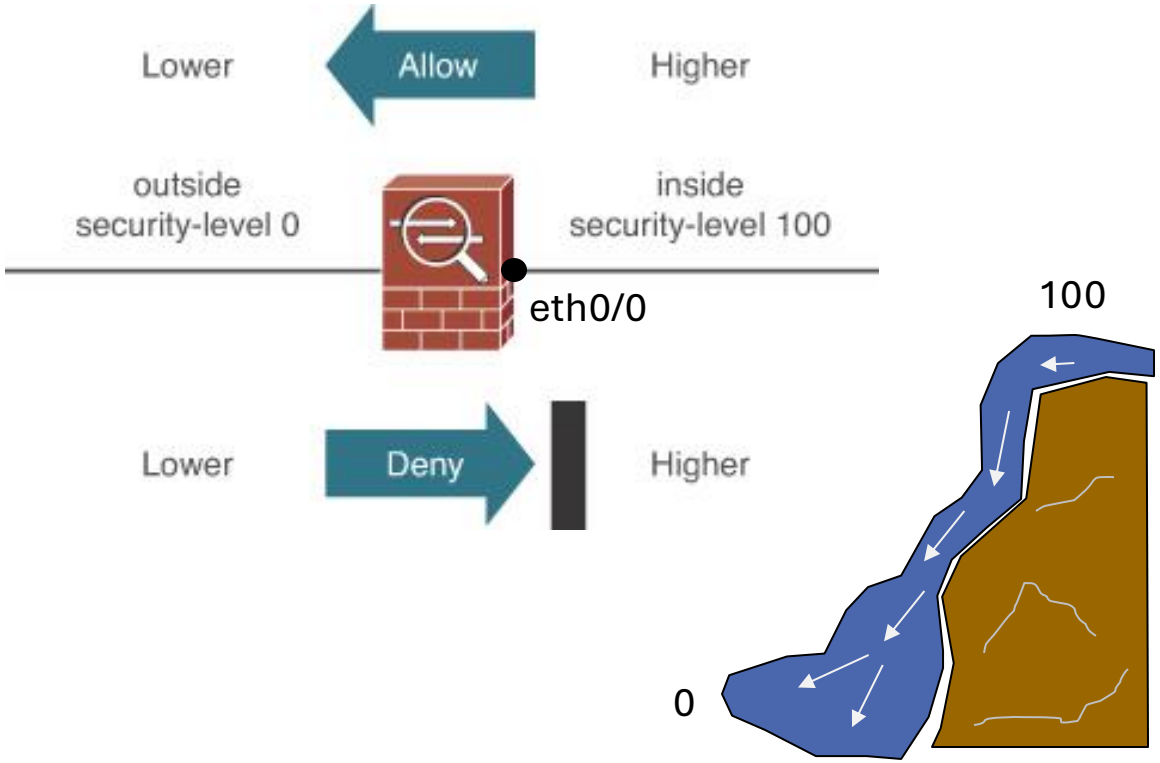


CISCO: ASA Firewall – Interfaces und Security Levels

- Dem externen Interface („outside“) und dem internen Interface („inside“) ordnet man unterschiedliche Sicherheits-Level zu.
- Die Sicherheits-Level können von 0 (das geringste Maß an Vertrauen) bis 100 (das größte Maß an Vertrauen) reichen.

Wasserfallmodell: Netzwerkpakete dürfen standardmäßig von hohen Werten zu niedrigen Werten fließen.

- Die Sicherheits-Level müssen pro Interface einen eindeutigen Wert haben.
- Die CISCO ASA lässt per Grundkonfiguration nur Datenverkehr vom Interface mit dem hohen Sicherheits-Level zum Interface mit dem niedrigen Sicherheits-Level zu.



- Sicherheits-Level können über den folgenden Befehl einem Interface zugeordnet werden

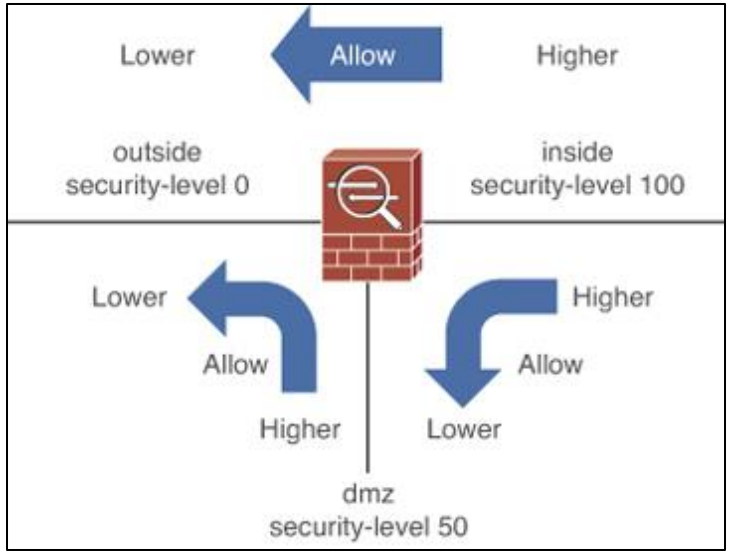
$$\text{level} \in \{0,1,\ldots,100\}$$

```
ciscoasa(config)# int eth0/0
ciscoasa(config-if)# security-level 100
```

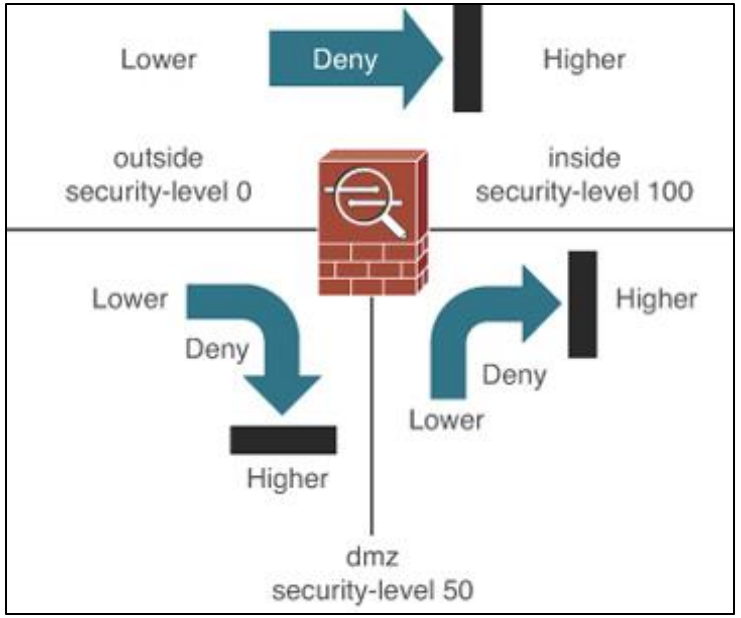
CISCO: ASA Firewall – 3 Interfaces und Security Levels

- Security-Levels können für mehrere Interfaces vergeben werden.
- Die Bilder zeigen den Fall, das zusätzlich eine DMZ (Demilitarisierte Zone) an die Firewall angeschlossen wird, der man den Sicherheits-Level 50 zugeordnet hat.
- Der folgende Netzwerkverkehr ist dann per Default erlaubt:
  - von Innen in die DMZ: $100 \rightarrow 50$
  - von Innen ins Internet: $100 \rightarrow 0$
  - von der DMZ ins Internet: $50 \rightarrow 0$
- Der restliche Verkehr ist per Default-Einstellung verboten.

Allowed

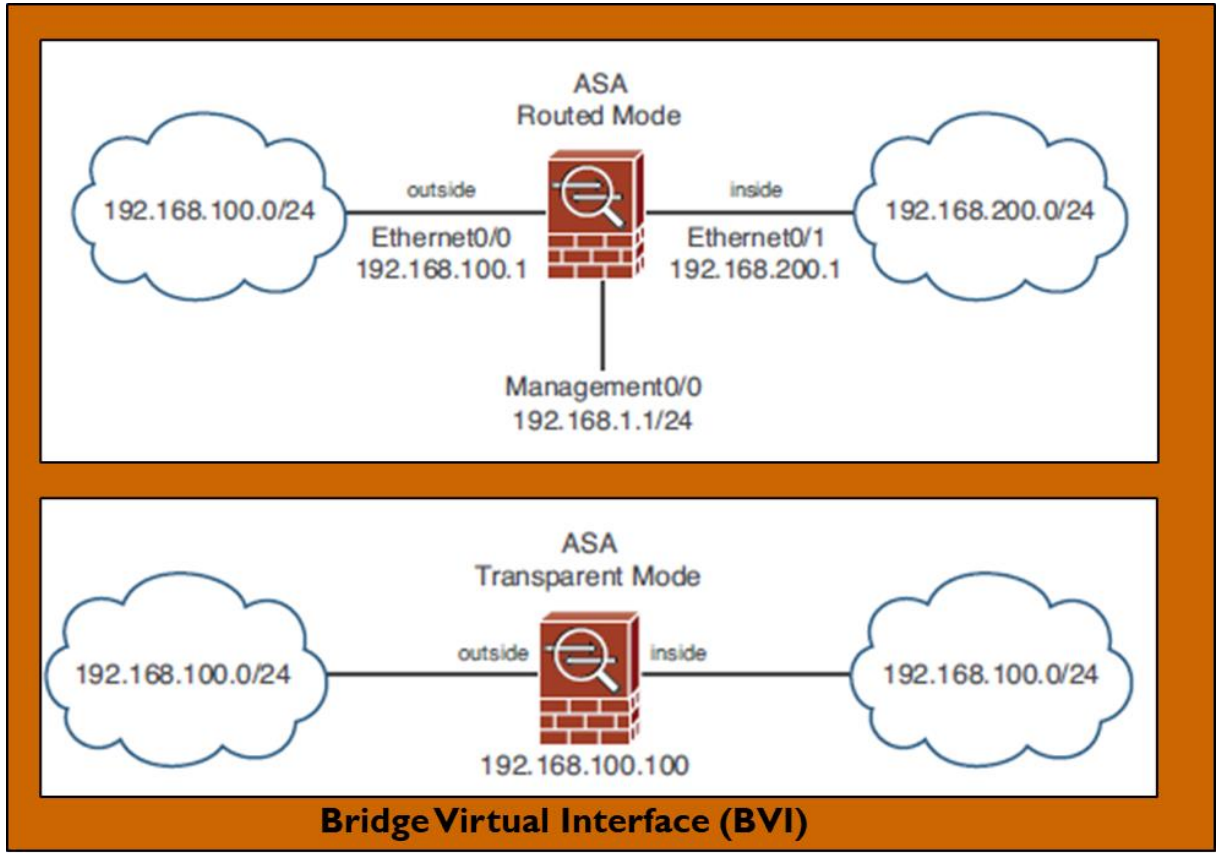


Denied



Firewall Modi: Routed oder Transparent

- Firewalls unterstützen prinzipiell zwei Modi:
  - den Routed Firewall-Modus
  - den Transparent Firewall-Modus
- Im Routed Mode arbeitet die Firewall als Router zwischen den beiden Netzwerken inside und outside. Die angeschlossenen Subnetze haben verschiedene Subnetz-Adressen. Firewall stellt das Gateway of last resort für das interne Subnetz dar.
- Im Transparent Mode arbeitet die Firewall als Bridge. Frames werden auf Basis der Layer2-Information weitergeleitet. Das interne und externe Interface werden zu einer Bridge-Group zusammengefasst. Der Bridge Group kann ein virtuelles IP-Interface zum Management der FW zugeordnet werden. Die Bridge-Group kennt kein "innen" und "außen".



Vergleich zwischen den beiden Modi

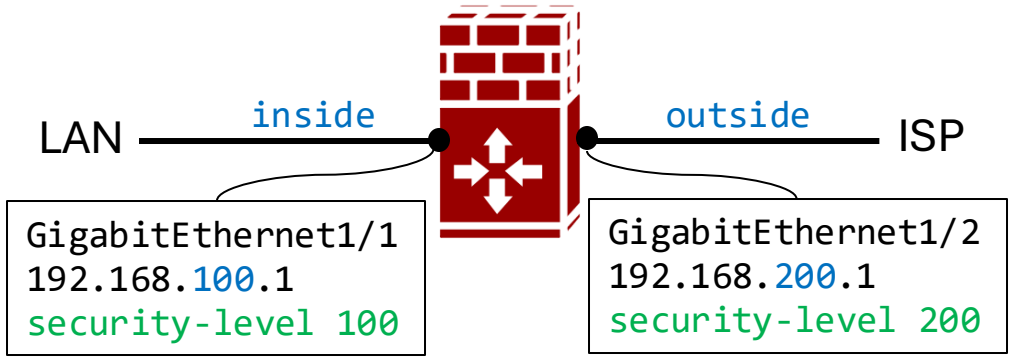
Eigenschaft	Routet Firewall Mode	Transparent Firewall Mode
Funktionsweise	**Filterung** von Netzwerk-Paketen **zwischen IP Subnetzen** (typisch: Innen und Außen).	**Filterung** von Netzwerk-Paketen innerhalb eines Layer-2-Netzwerkes.
Layer	**Layer-3**: Jedes Interface muss sich in einem separaten Subnetz befinden und benötigt eine IP-Adresse.	**Layer-2**: Es werden keine Subnetze und keine IP-Adressen benötigt.
Routing	Unterstützt dynamisches und statisches IP-Routing und NAT	Kein Routing, kein NAT nur Layer-2 Weiterleitung
EInsatzszenario	Schutz eines internen Subnetzes mit Routing-Funktionalität in andere interne Subnetze. Perimeter-Firewall ins Internet	Netzwerk-Segmentierung auf Layer-2mit einer Firewall die inline zwischen 2 Switche geschaltet wird, ohne die IP-Topologie zu ändern.
Beispiel	Perimeter-Firewall ins Internet	Firewall in der DMZ zur Segmentierung der DMZ.

Konfiguration einer CISCO ASA im Routed Mode

- Jedem ASA-Interface muss neben dem Security-Level auch ein Interfacename und eine statische IP-Adresse zugeordnet werden, inklusive der zugehörigen Subnetzmaske.
- Im Beispiel erhält das innere Interface den Security-Level 100 und das äußere Interface den Security-Level 0.

```
ciscoasa(config)# interface GigabitEthernet1/1
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 192.168.100.1 255.255.255.0
ciscoasa(config-if)# no shutdown
```

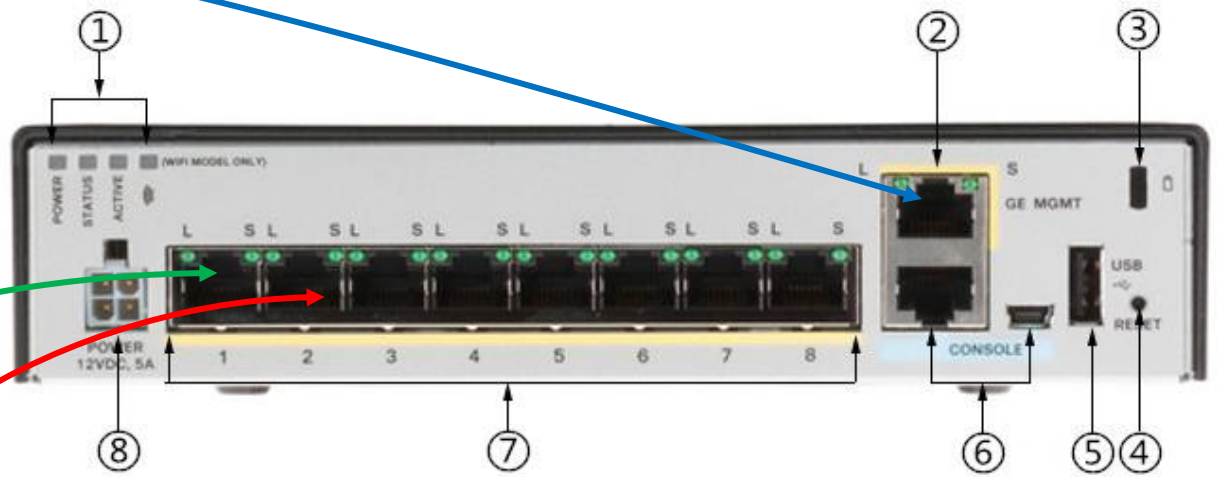
```
ciscoasa(config)# interface GigabitEthernet1/2
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# security-level 0
ciscoasa(config-if)# ip address 192.168.200.1 255.255.255.0
ciscoasa(config-if)# no shutdown
```



CISCO ASA 5506 (Labor): Default Konfiguration

- ASA 5506 besitzt 8 Gigabit-Anschlussports.
- Besitzt einen expliziten Management-Port.
- Interface G1/1 und G1/2 sind als internes und als externes Interface per Default vorkonfiguriert:

```
interface GigabitEthernet1/1
  nameif inside
  security-level 100
  ip address 192.168.1.1 255.255.255.0
!
 interface GigabitEthernet1/2
  nameif outside
  security-level 0
  ip address dhcp
```



①	Status LEDs	⑤	USB Port
②	Management port	⑥	Console ports
③	Lock slot	⑦	Network data ports
④	Reset button	⑧	Power cord socket

Firewall und Nat

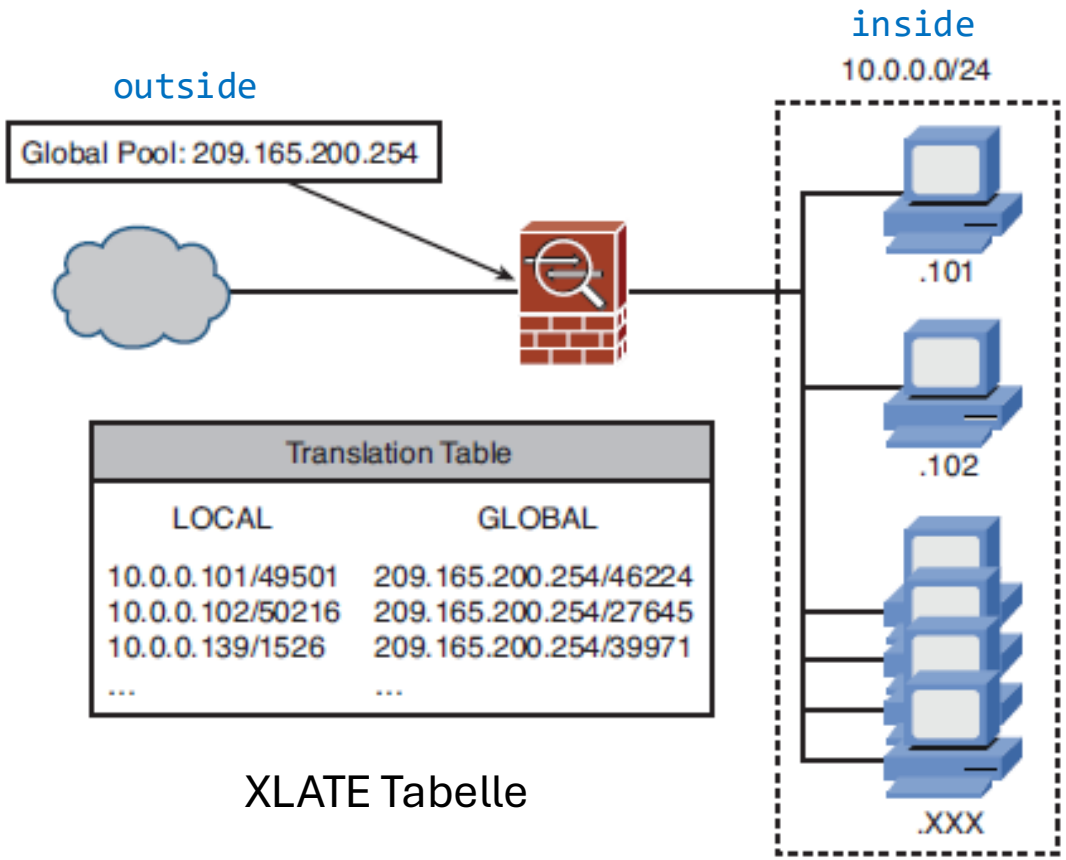
Damit Sie aus ihrem Unternehmensnetzwerk mit Systemen im Internet kommunizieren können, verwenden viele Firmen das Network Address Translation (NAT) - Verfahren, bei dem private IP-Adressen eines Unternehmens durch die Firewall in öffentliche IP-Adressen übersetzt werden:

(1) NAT verringert den Bedarf an öffentlichen IP-Adressen.

(2) Mit NAT kann ein Unternehmen einen Wechsel des Internetdienstanbieters (ISP) vornehmen, ohne seine internen, privaten IP-Adressen ändern zu müssen.

(3) NAT verbirgt das interne IP-Adressierungsschema vor dem öffentlichen Internet, was die Netzwerksicherheit erhöht.

(4) Beim Dynamic Inside NAT werden die Source-IP-Adressen der internen Hosts dynamisch einem Adress-Pool an öffentliche IP-Adressen und TCP-Portnummern zugewiesen.



XLATE Tabelle

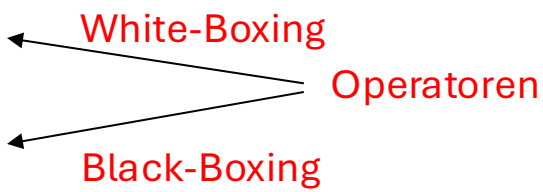
xlate ist die Kurzschrift von translate

Packet-Filter Firewalls

- Eine Packet-Filter-Firewall analysiert auf Basis eines Regelsatzes, die eingehenden und ausgehenden IP-Packete.
- Die Regeln werden in Form von Access-Control-Listen (siehe vorne) umgesetzt.
- Bei mehreren ACL-Regeln werden diese für jedes Packet sequenziell abgearbeitet: first-match Prinzip
- Bei Übereinstimmung mit der ACL-Regel, wird das Paket entweder

(1) weitergeleitet: permit

(2) verworfen: deny



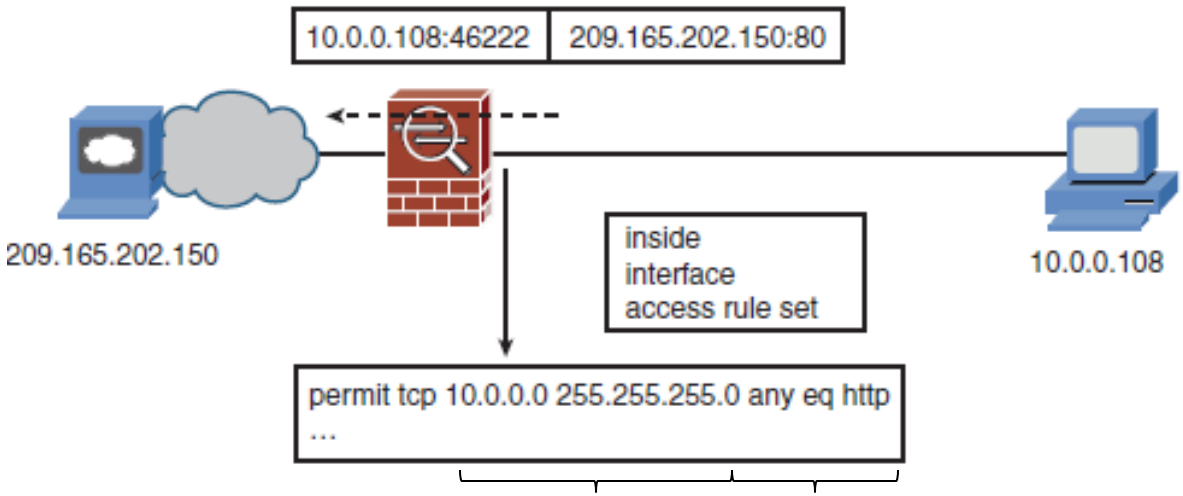
- Wenn keine Übereinstimmung mit einer Regel besteht, wird eine Standard-Aktion ausgeführt.
- Eine sichere Firewall sollte als Standard-Aktion „Packet verwerfen“ setzen:
  - Default-is-Deny Prinzip auch als Fail-Safe-Default bezeichnet.

Packet-Filter Firewalls

- ACL-Regeln basieren auf den folgenden Informationen, die in einem IP-Netzwerkpaket enthalten sind:
  - Quell-IP- und Ziel-IP-Adressen (IP-Header) im Paket
  - Verwendetes Upper-Layer-Protokoll (IP-Header UpperLayer Feld): UDP,TCP,ICMP
  - Quell-Port- und Ziel-Portnummer (TCP-Header) im Paket

    Für die Portnummern dürfen auch "Portnamen" verwendet werden:

    80 – http , 443 – https, 53 - dns, 22 - ssh, 25 – smtp, ...
- Beispiel: Alle internen Clients dürfen im Internet nur verschlüsselt browsen:



Source: Interne Clients im Subnetnetz 10.0.0.0/24

Destination: Alle Hosts, Portnummer=80 (http)

Upper-Layer Protocol

beliebiger Ziel-Host mit Port 443

```
#access-list extended LAN_to_Internet permit tcp 10.0.0.0 255.255.255.0 any eq https
```

(2.1)

Name der ACL

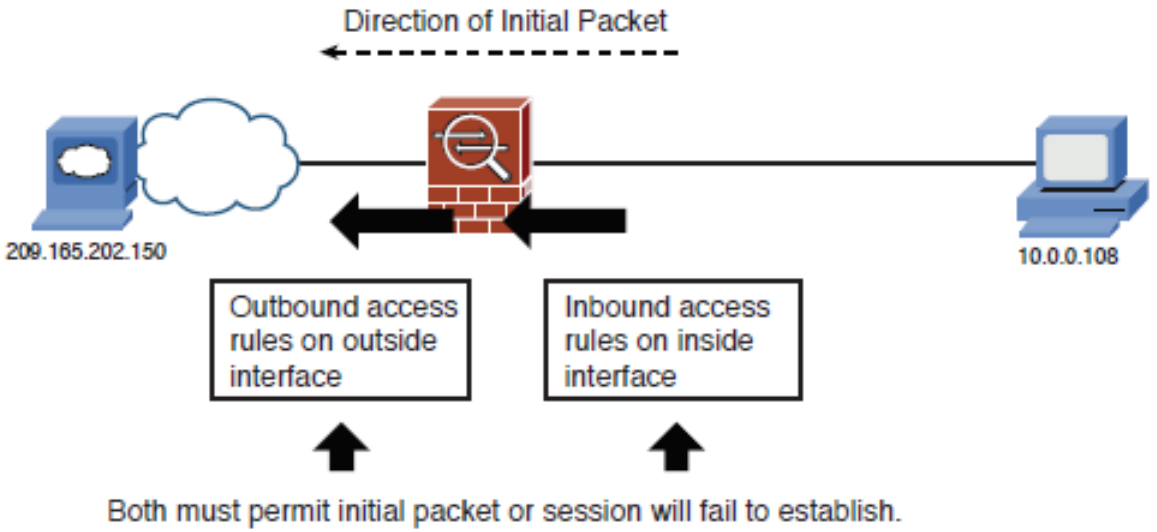
Source-Netzwerk

Packet-Filter Firewalls

- Wie schon bei Routern gezeigt, muss die ACL anschließend einem Firewall Interface zugewiesen werden und die Fluss-richtung des Netzwerkverkehrs angegeben werden.
- Dies geschieht mit dem Befehl `access-group`.
- Beispiel:

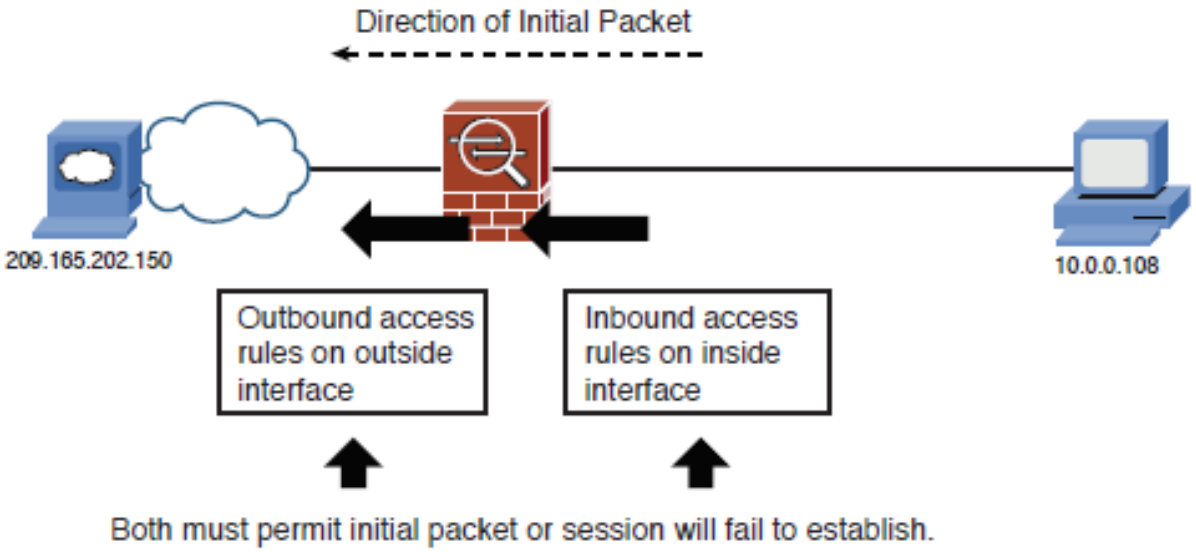
```
#access-group LAN_to_Internet in interface inside
```

  - ACL mit dem Namen `LAN_to_Internet` wird dem Interface `inside` der Firewall zugewiesen.
  - Inbound (in): Die ACL wird auf Datenverkehr angewendet, der über die angegebene Schnittstelle `inside` in die Firewall eintritt.
  - Outbound (out): Die ACL wird auf den Datenverkehr angewendet, der die Firewall über die angegebene Schnittstelle `inside` verlässt.



Inbound vs. Outbound Access-Listen

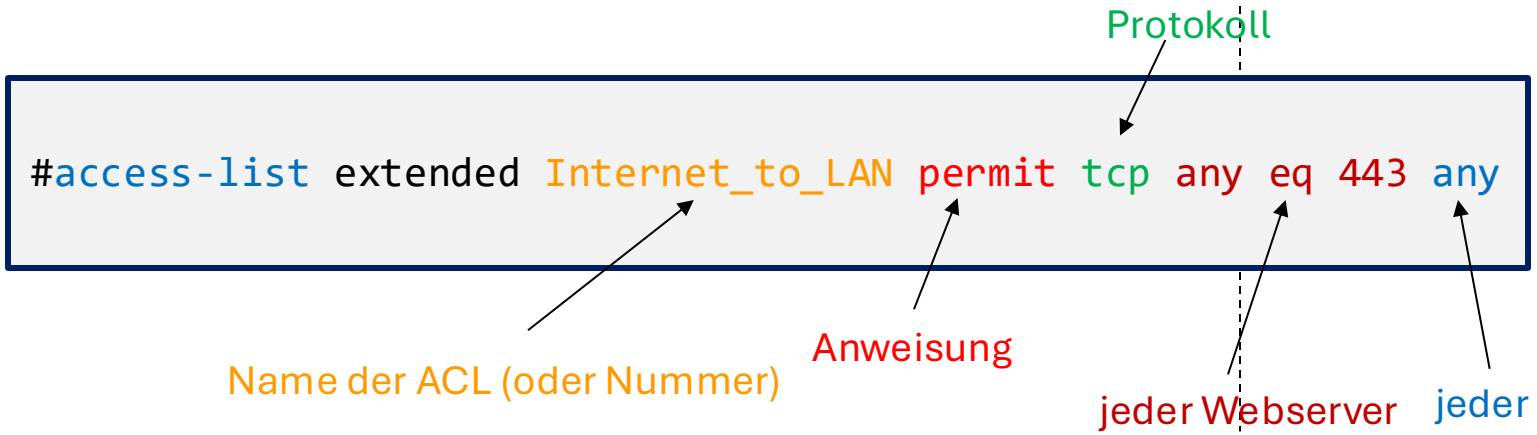
- Inbound-Access-Liste
  - Vor der Verarbeitung eines Paketes durch den Routing Prozess wird dieses gegen die Inbound-Access-Liste geprüft ob es gelöscht oder weitergeleitet werden soll: zuerst prüfen, dann routen
  - Da Routing im zweiten Schritt erfolgt, ist eine Inbound-Access Liste am internen Interface für Datenverkehr aus dem Unternehmen effizienter als eine Outbound-Access Liste.
- Outbound-Access-Liste
  - Nach der internen Weiterleitung durch den Routing-prozess und unmittelbar vor Versendung des Paketes am entsprechenden Outbound Router Interface, wird das Paket gegen die Outbound-Access-Liste geprüft.

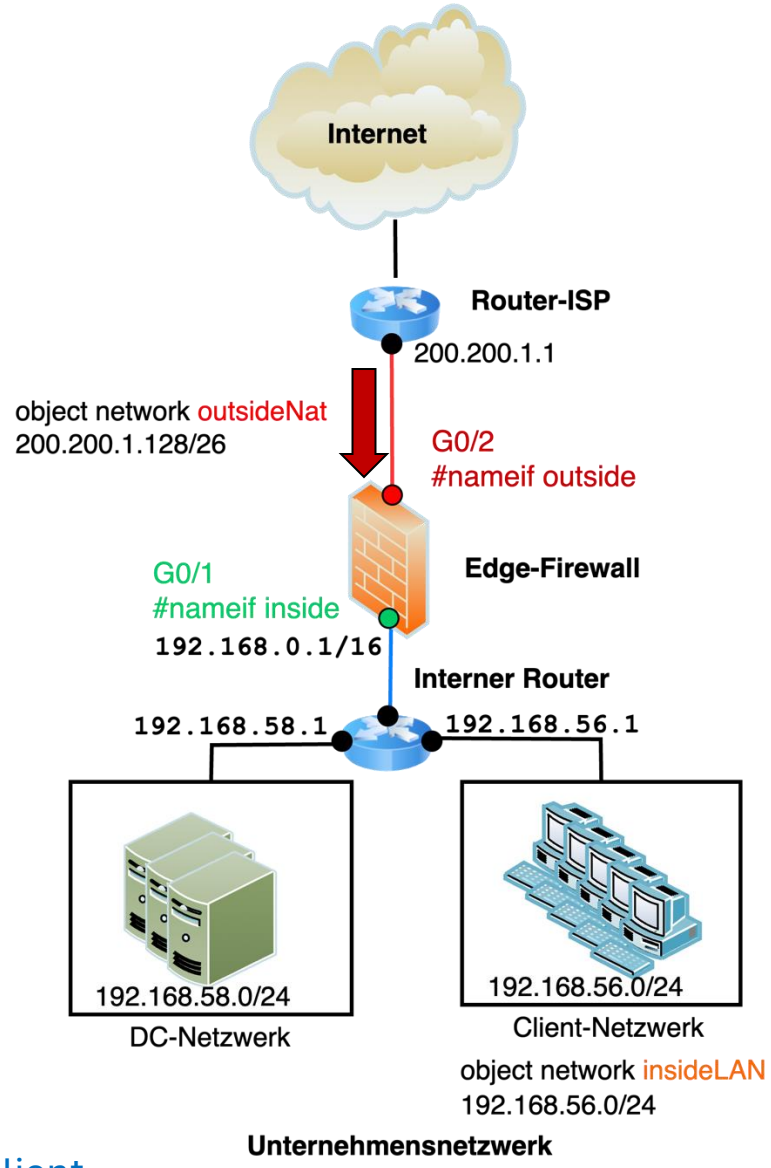


- Ein Paket wird nur zugestellt, wenn die Inbound- und die Outbound-ACL es erlauben.

Beispiel: CISCO ASA ACL – Teil 1

- Klassische Konfiguration Teil1: Damit eine Web-Client im Internet browsen kann, muss der Antwortdatenverkehr von Webservern zugelassen werden.
  - Verkehr von Inside nach Outside ist durch die vorherige ACL-Whitelist (2.1)erlaubt.
  - Dynamisches Inside NAT: nur Anfragen, die von Innen initiiert werden sind über NAT-Tabelle beantwortbar.
  - Eingehender Antwort-HTTP- Datenverkehr von Außen (outside) nach Innen (inside) muss explizit freigeschaltet werden:

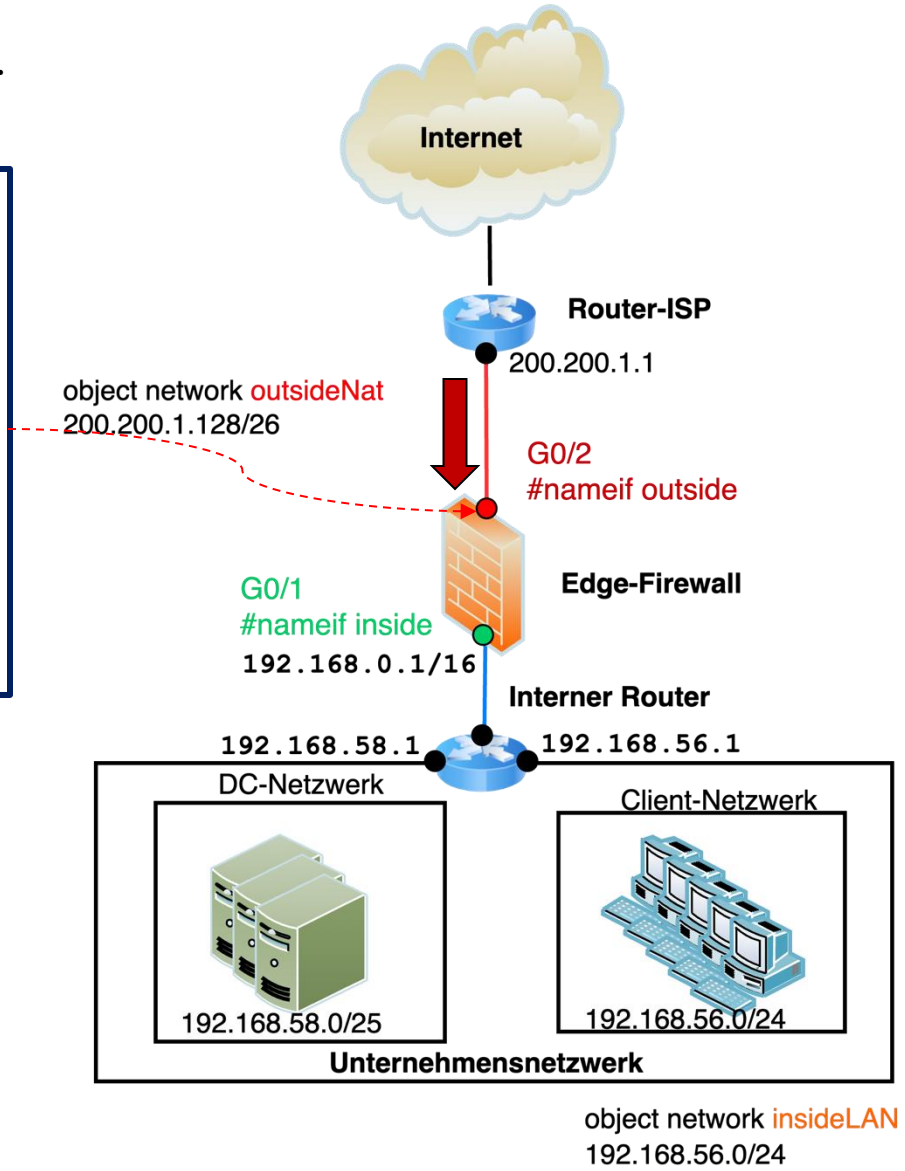




Beispiel: CISCO ASA ACL-Teil2

- Klassische Konfiguration Teil2: Interne Clients sollen mittels ICMP die Netzwerkverbindung zu Hosts im Internet überprüfen können. Umgekehrt aber nicht.

```
!Erlaube ICMP-Antwortverkehr (echo-reply) vom Sender im Internet
(Jeder) zum internen Client (Jeder)
#access-list extended Internet_to_LAN permit icmp any echo-reply any
!Wende ACL am outside-Interface für Verkehr von Außen nach Innen
(ingress) an
#access-group Internet_to_LAN in interface outside
```



Bedeutung der Parameter:

echo-reply: ICMP Echo Reply als Antwort auf einen ICMP Echo Request

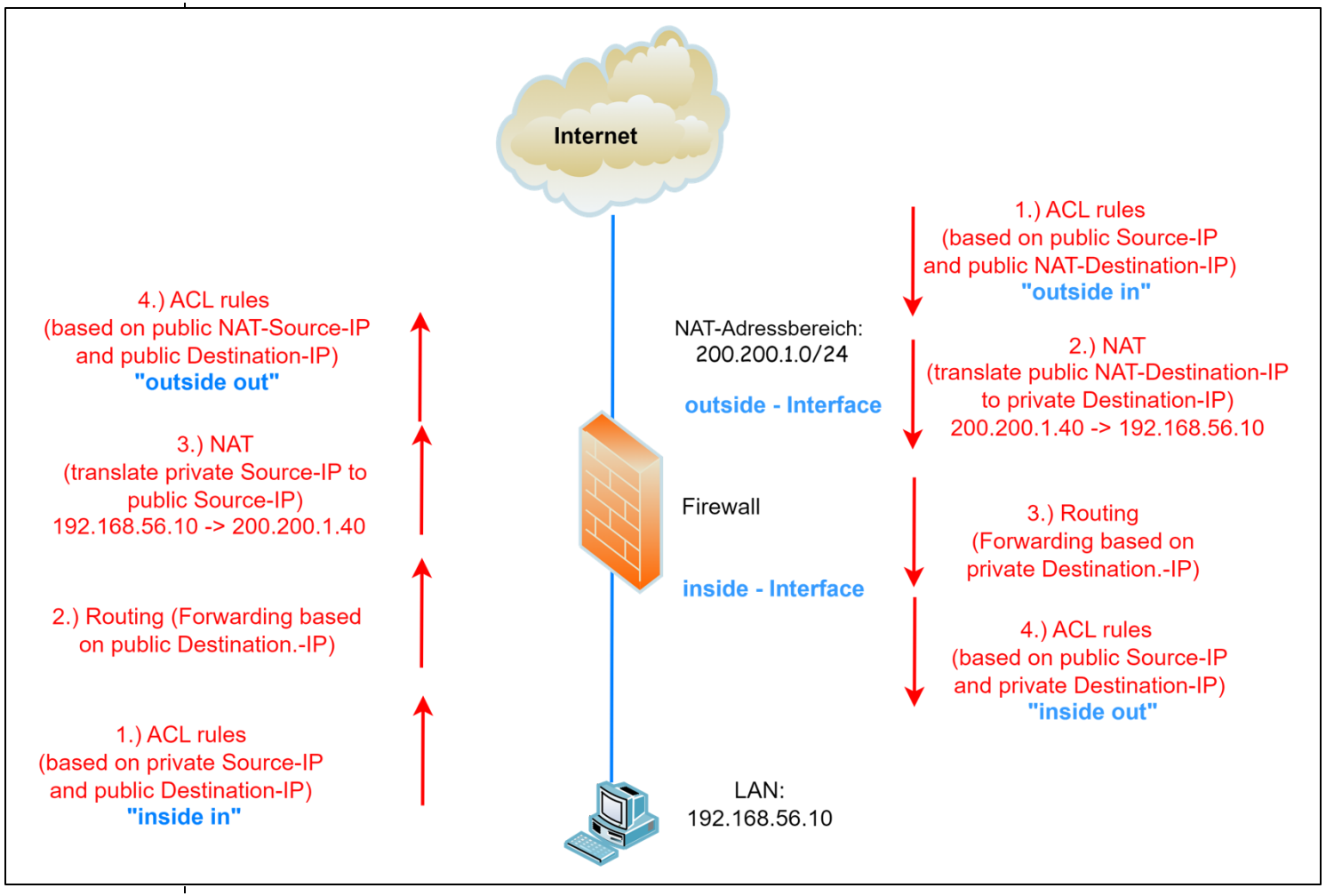
access-list Internet_to_LAN : ACL mit Bezeichner „Internet_to_LAN“

access-group : anwenden der ACL an einem Interface unter Angabe der Flussrichtung.

in: Ingress Datenverkehr (eingehender Datenverkehr am outside Interface)

Order of Operation: ACL, NAT, Routing

- Inside In und Outside In ACLs werden immer zuerst durchgeführt: eingehender Verkehr
- Inside Out und Outside Out ACLs werden immer zuletzt durchgeführt: ausgehender Verkehr
- Packete die durch das Inside-NAT-Interface in die Firewall gelangen, werden zuerst geroutet und dann genattet.
- Packete die durch das Outside-NAT-Interface in die Firewall gelangen, werden zuerst genattet und dann geroutet.



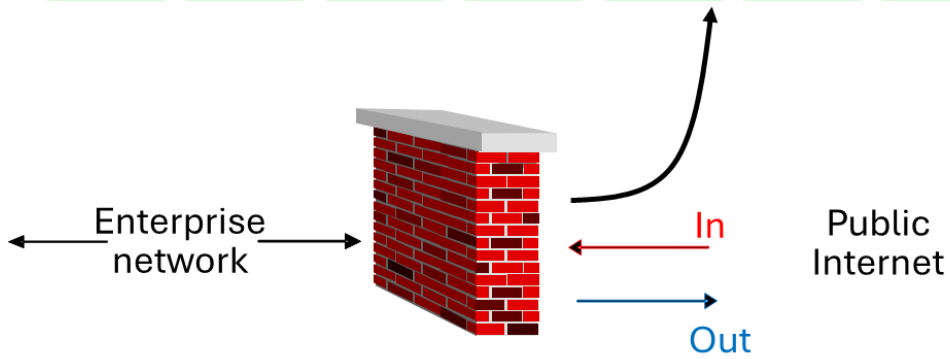
Grenzen einer Packet-Filter Firewall – Teil 1

Szenario: Regelsatz für E-Mailing (SMTP: TCP/25) in einem Unternehmen

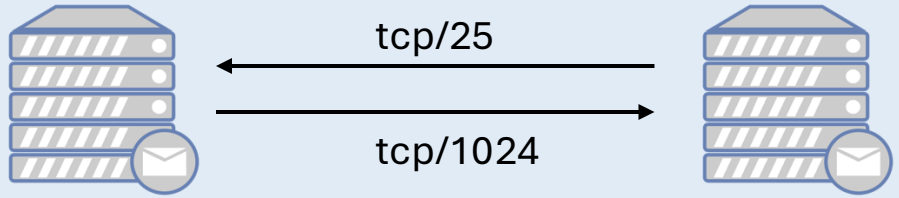
- Rule A: Eingehende E-Mail und damit eingehende TCP-Verbindung von externem Mail-Server an interne IP-Adresse mit Port 25.
- Rule B: Ausgehende TCP-Verbindung auf TCP-Port > 1023 um eingehende E-Mails zu bestätigen.

ACLs am externen Interface

Rule	Direction	Src address	Dest addresss	Protocol	Dest port	Action
A	In	External	Internal	TCP	25	Permit
B	Out	Internal	External	TCP	>1023	Permit
C	Out	Internal	External	TCP	25	Permit
D	In	External	Internal	TCP	>1023	Permit
E	Either	Any	Any	Any	Any	Deny



internes Interface externes Interface



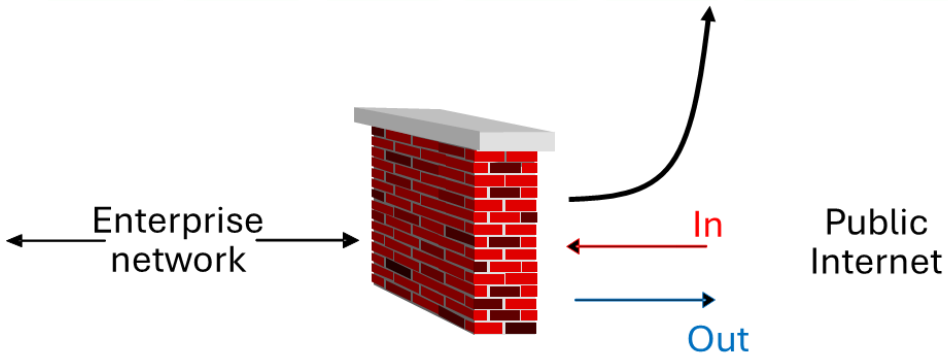
Grenzen einer Packet-Filter Firewall – Teil 1

Szenario: Regelsatz für E-Mailing (SMTP: TCP/25) in einem Unternehmen

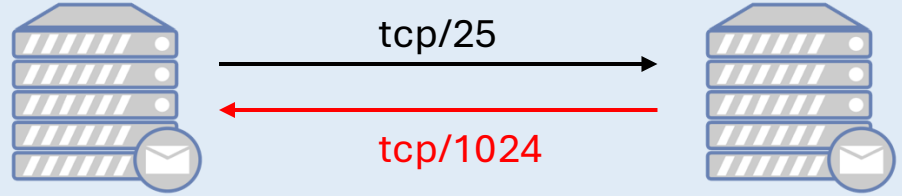
- Rule C: Ausgehende SMTP-Verbindung von internem Mail-Server zu externem Mail-Server auf TCP-Port 25.
- Rule D: Bestätigung ausgehender SMTP-Verbindung durch externen SMTP-Client mit TCP-Port > 1023.
- Rule E: Restlicher Verkehr wird verboten (implizit)
- Problem: Rule D in ACL-Liste ermöglicht beliebigen Netzwerkverkehr von außen nach innen auf Ports > 1023 zum internen E-Mail-Server.

ACLs am externen Interface

Rule	Direction	Src address	Dest addresss	Protocol	Dest port	Action
A	In	External	Internal	TCP	25	Permit
B	Out	Internal	External	TCP	>1023	Permit
C	Out	Internal	External	TCP	25	Permit
D	In	External	Internal	TCP	>1023	Permit
E	Either	Any	Any	Any	Any	Deny



internes Interface externes Interface



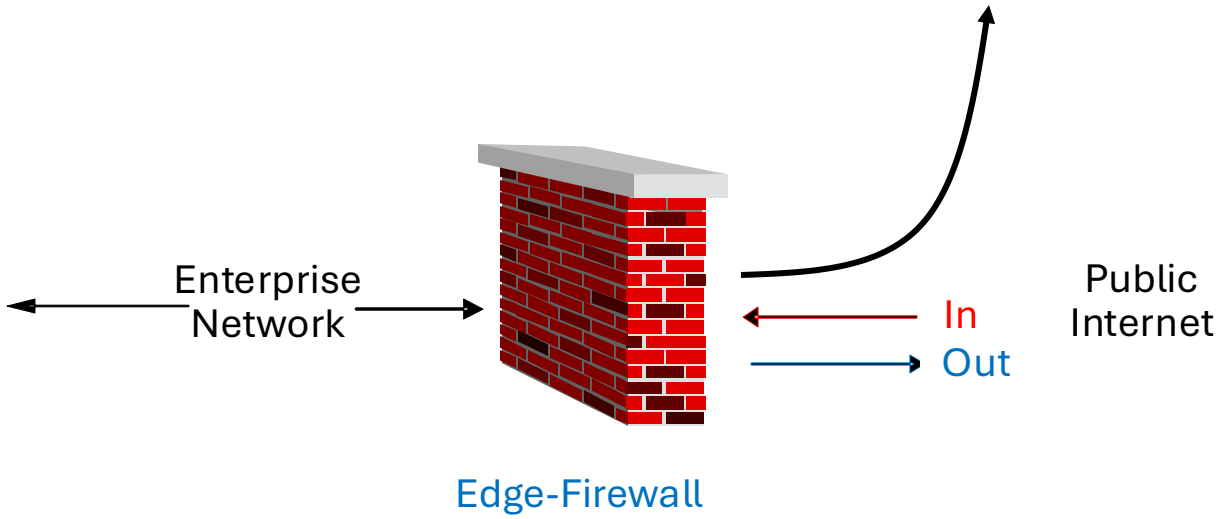
Grenzen einer Packet-Filter Firewall – Teil 2

- Wie kann Rule D verbessert werden, sodass die ursprüngliche Idee nur die Bestätigung eines externen SMTP-Clients zuzulassen, erfüllt wird?
- Idee: Rule D um TCP-Flags erweitern
  - Eingehendes Packet muss das TCP-ACK-Flag (ACK=1) gesetzt haben und die richtige ACK-Nummer (ISN+1) enthalten

    → Stateful Inspection Firewall
  - Einsatz von Dynamischen Inside NAT

    → nur Verbindung möglich, wenn vorher der SMTP-Server eine Verbindung nach außen initiiert hat, die zu einem Eintrag des SMTP-Servers in der NAT-Tabelle führt.

Rule	Direction	Src address	Dest addresss	Protocol	Dest port	Flag	Action
A	In	External	Internal	TCP	25		Permit
B	Out	Internal	External	TCP	>1023		Permit
C	Out	Internal	External	TCP	25		Permit
D	In	External	Internal	TCP	>1023	ACK	Permit
E	Either	Any	Any	Any	Any		Deny



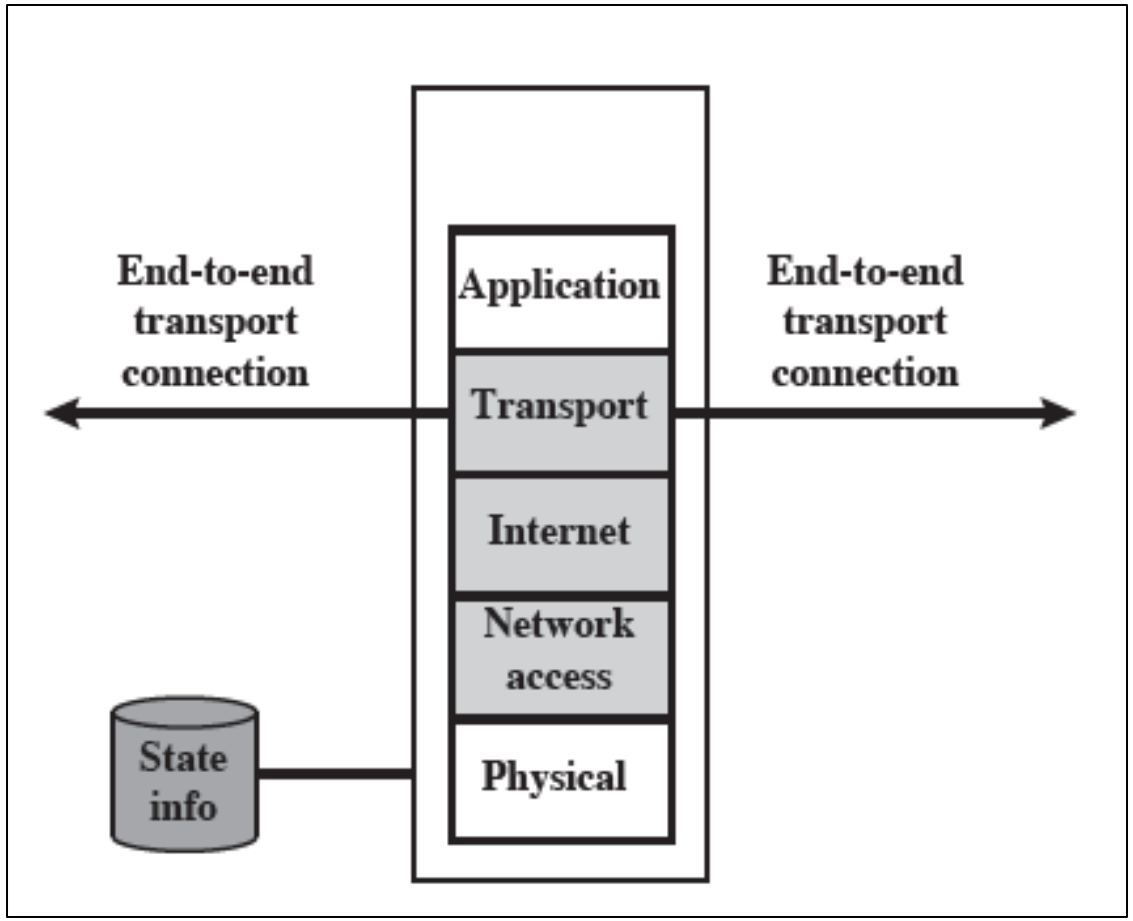
Stateful Inspection Firewall

- Ankommende Packet werden zuerst überprüft, ob
  - eine bestehende Verbindung in der Status-Verbindungs-Tabelle vorhanden ist und wenn ja,
  - ob die Protokollstatusinformation mit dem Eintrag in der Status-Verbindungs-Tabelle konsistent ist.
- Das Überprüfungsergebnis ergibt 3 mögliche Ergebnisse:

a) Beides trifft zu: Das Packet wird von der Firewall durchgelassen.

b) Verbindung vorhanden, Packetstatusinformation nicht konsistent: Das Packet wird verworfen.

c) Keine Verbindung vorhanden: Anwenden der ACL-Regeln (Default-is-Deny)



Status Tabelle: Inhalte

Protokoll	Zusatzinformation
IP, TCP	Quell- und Zieladresse, Quell- und Zielports TCP-Flags (SYN, ACK,FIN,RST) Sequenz- und ACK-Nummern von Sender und Empfänger IDLE-Time
IP, UDP	Quell- und Zieladresse, Quell- und Zielports, IDLE-Time
IP, ICMP	Quell- und Zieladresse, ICMP-Type und Code, ICMP Identifier, ICMP Sequencenumber IDLE-Time

```
!Anzeige der Status Tabelle auf einer CISCO ASA

#show conn

!Löschen von Einträgen in der Status Tabelle einer
CISCO ASA

#clear conn address 192.0.2.146
```

Status-Tabelle: Idle-Time

- Die Idle-Time ist ein Parameter, um die Aktualität der Einträge in der Statustabelle zu sichern.
- Wenn eine Sitzung inaktiv wird d. h. keine Daten gesendet oder empfangen werden, beginnt die Firewall, den Idle-Timer zu starten.
- Bleibt die Verbindung länger inaktiv als ein vorkonfigurierter Timeout-Wert, entfernt die Firewall die Verbindung aus der Statustabelle.
- Nachfolgende Pakete, die eventuell zu dieser Verbindung gehören, werden als neu betrachtet und müssen den Verbindungsaufbauprozess erneut durchlaufen.

```
! Set TCP idle timeout to 2 minutes

ciscoasa(config)# timeout conn 2:00

! Set UDP idle timeout to 2 minutes

ciscoasa(config)# timeout udp 2:00

! Set ICMP idle timeout to 30 seconds

ciscoasa(config)# timeout icmp 0:00:30
```

Connection State Table

- Client (172.16.2.23) im Unternehmensnetzwerk kontaktiert FTP-Server (192.168.1.34) außerhalb. Eintrag in Status-Tabelle mit dem Status "SYN_SENT":

Protokollnummer für TCP in IP-H

TCP-Status

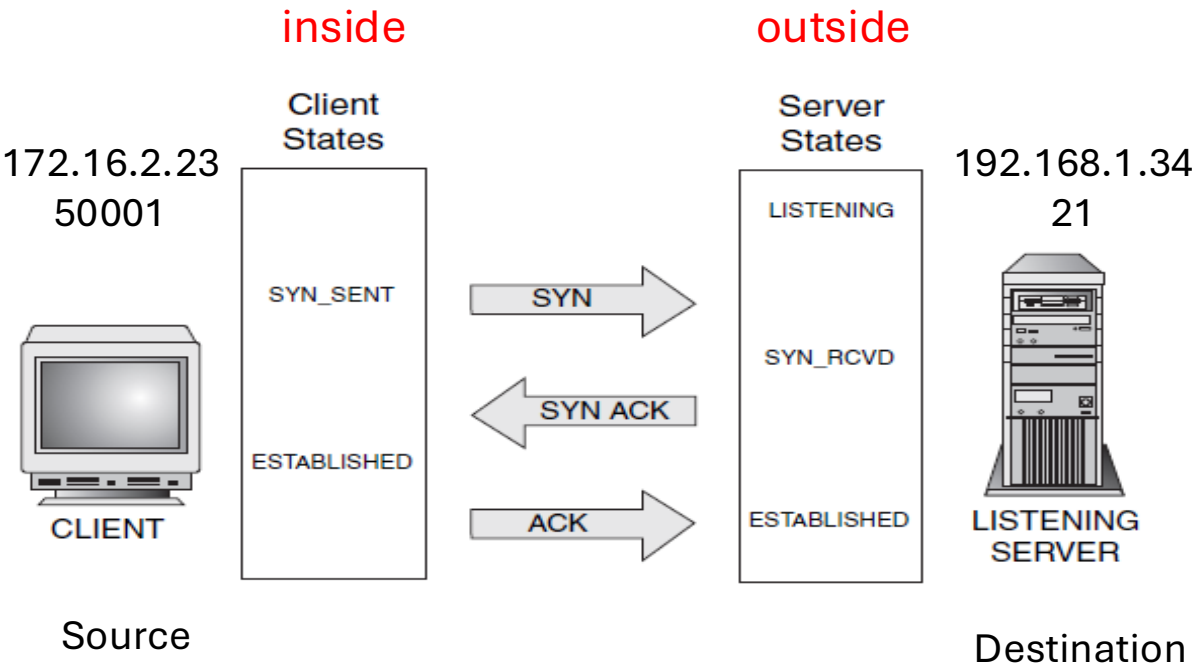
```
tcp 6 11 SYN_SENT outside=192.168.1.34
inside=172.16.2.23 srcport=50001 dstport=21
[UNREPLIED]
```

- Nach der Verbindungbestätigung durch den Server geht der Status in den Zustand „ESTABLISHED“ über

Idle-Timer: Zeit für die der Eintrag noch gültig ist

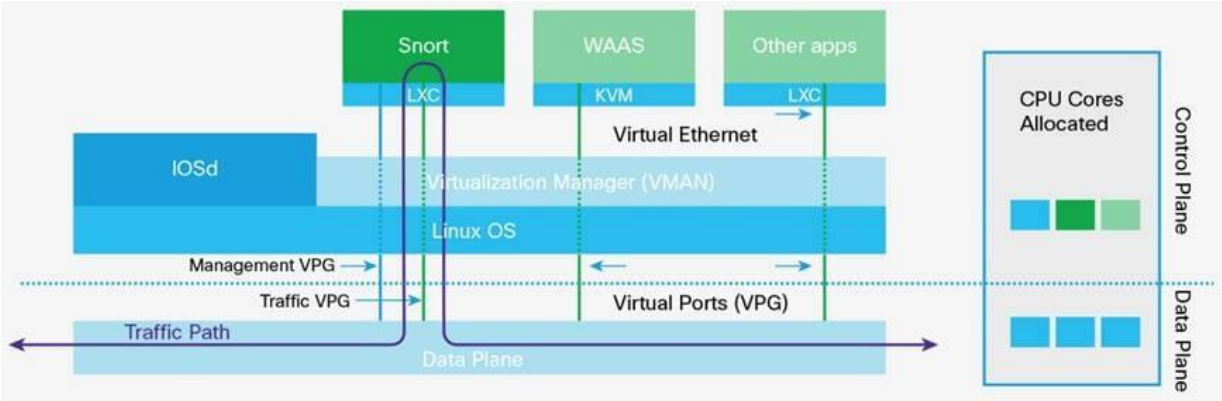
```
tcp 6 7 ESTABLISHED outside=192.168.1.34
inside=172.16.2.23 srcport=50001 dstport=21
```

TCP-Verbindungsaufbau



Next Generation Firewalls

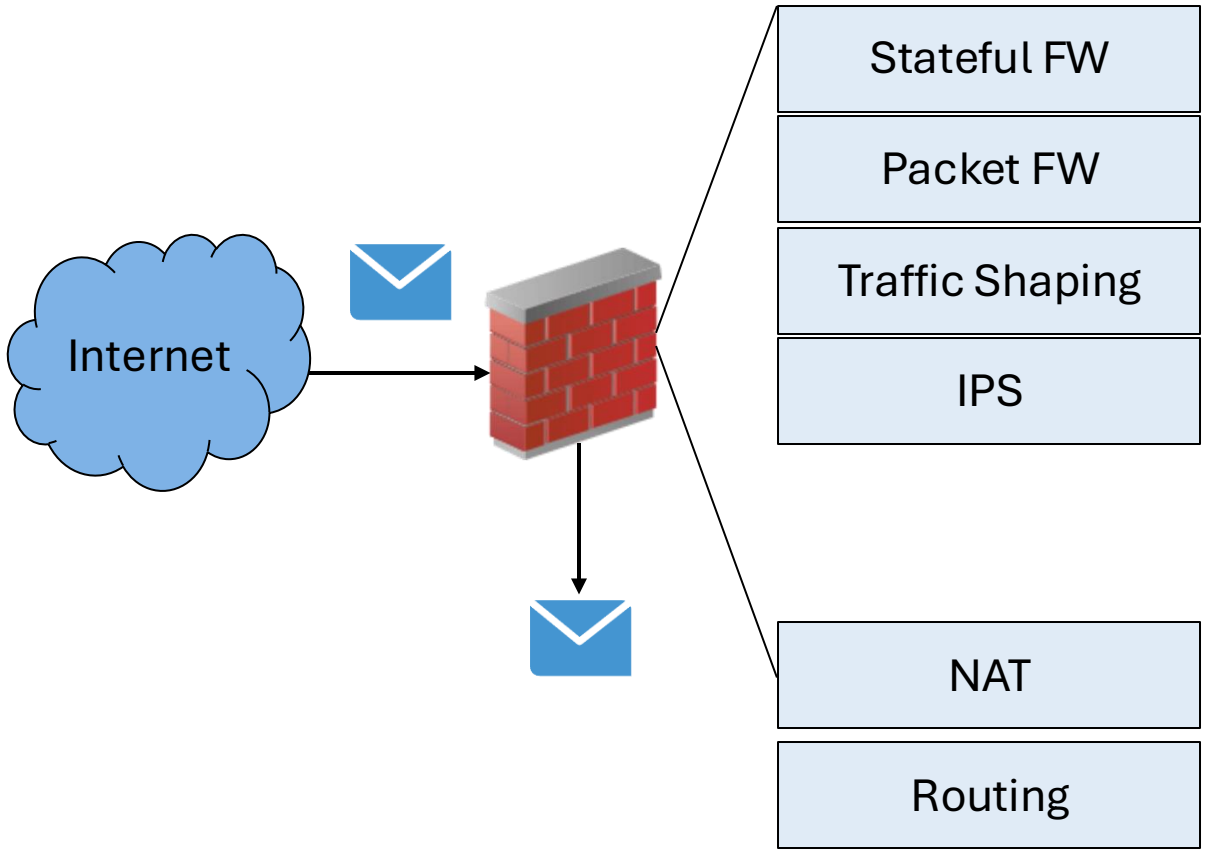
- Eine Next Generation Firewalls (NGFW) ist modernes Firewall-System, das neben den Fähigkeiten einer Stateful Firewall weitere Systeme wie z.B. ein Intrusion Protection System (IPS) und weitere Security-Appliances wie z.B. Sandboxes, Anti-Malware- und Anti-Spam-Systeme enthält.
- Die zusätzlichen Systeme werden als eigenständige virtuelle Container unter Einsatz von LXC (Linux Container) auf einer CISCO ASA betrieben.



- NGFW arbeitet somit als Deep Package Inspection (DPI) System und kann den eingehenden Datenverkehr klassifizieren und filtern kann, auf Basis aller Protokollschichten:
  - Verkehrsklassifizierung: Identifizieren und Priorisieren von Anwendungen und Protokollen.
  - Bandbreitenverwaltung: Verwalten und Begrenzen der Bandbreite in Abhängigkeit der Daten.
  - Inhaltsfilterung: Blockieren unerwünschter oder nicht konformer Inhalte.
  - Anwendungskontrolle: Zulassen oder Blockieren von Anwendungen basierend auf Richtlinien.

Arbeitsweise einer NGFW

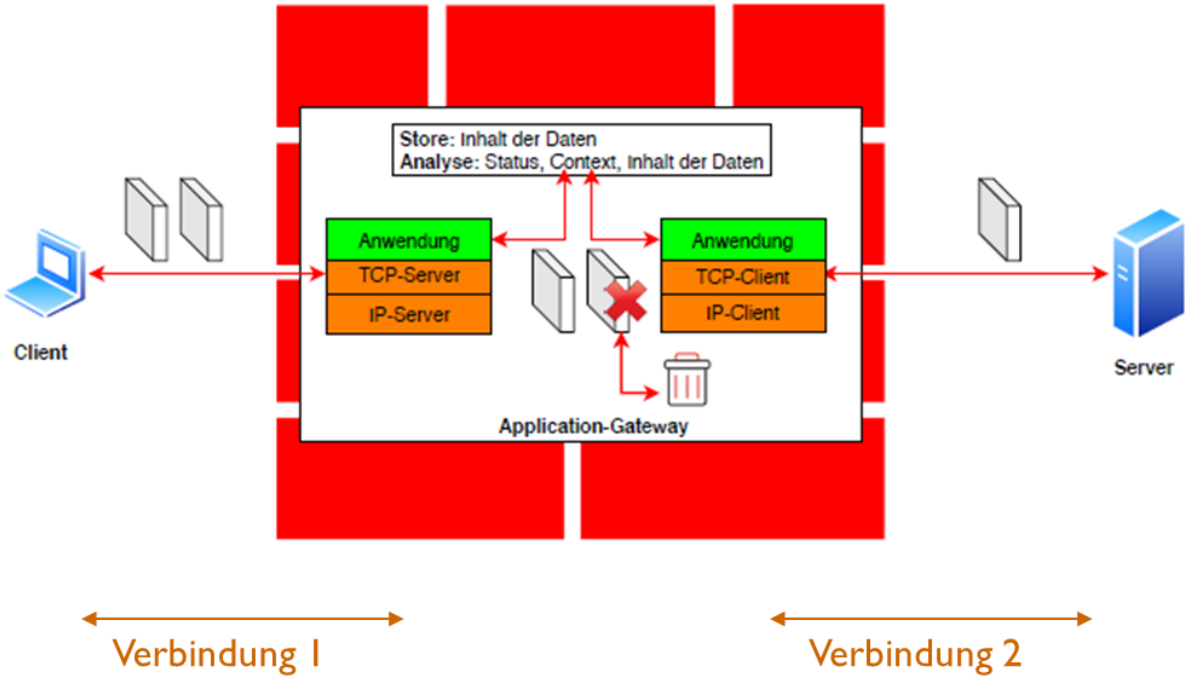
- Inbound Datenverkehr am Outside-Interface (vom Internet ins Unternehmenswerk) wird innerhalb der NGFW sukzessive analysiert
  - (1) Stateful FW: Verbindungsstatus überprüfen
  - (2) Packet-FW: Access-Control-Regeln anwenden (Packet-FW)
  - (3) Traffic Shaping: Bandbreitenmanagement (Priorisierung oder Drosselung von Applikationsverkehr)
  - (4) IPS-System: Filtern von Schadverkehr
  - (5) NAT-Rückübersetzen
  - (6) Routing



2.2 Application-Firewalls und DMZ

Ziele, Konfiguration, Ausprägungen,

Platzierung, Beispiele für APP-FW

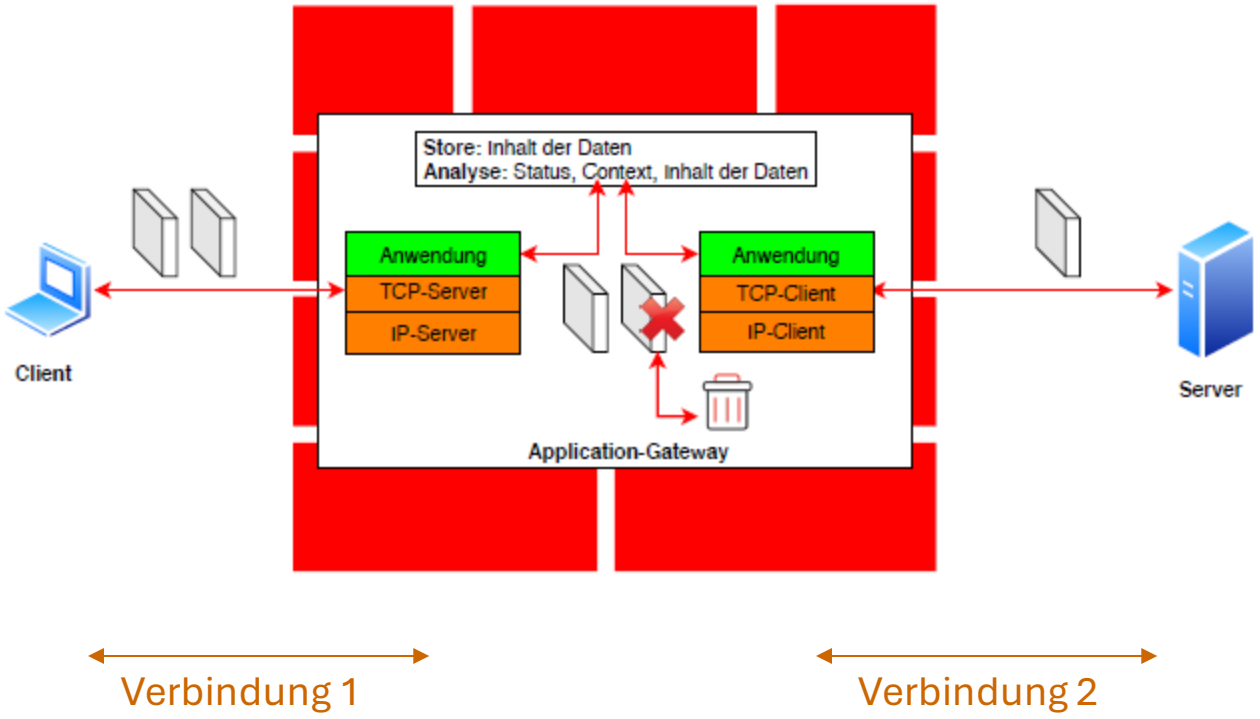


Application Layer Firewalls – Application Gateway

- Erfolgt die Filterung der Packete auf Basis eines spezifischen Anwendungsprotokolls (z.B.: HTTP, SMTP), so bezeichnet man die Firewall als Application Gateway.
- Das Application Gateways schaltet sich dabei als Proxy zwischen den Client und Server und kann so den Datenverkehr vollständig kontrollieren
  - SMTP: Secure-E-Mail-Gateway, SPAM-Filter
  - HTTP: Web-Proxy mit Content-Filter

Ein Proxy ist ein zwischengeschalteter Agent (SW oder SW/HW), der im Namen eines Endpunkts agiert, ohne eine direkte Verbindung zwischen den beiden Endpunkten zuzulassen.

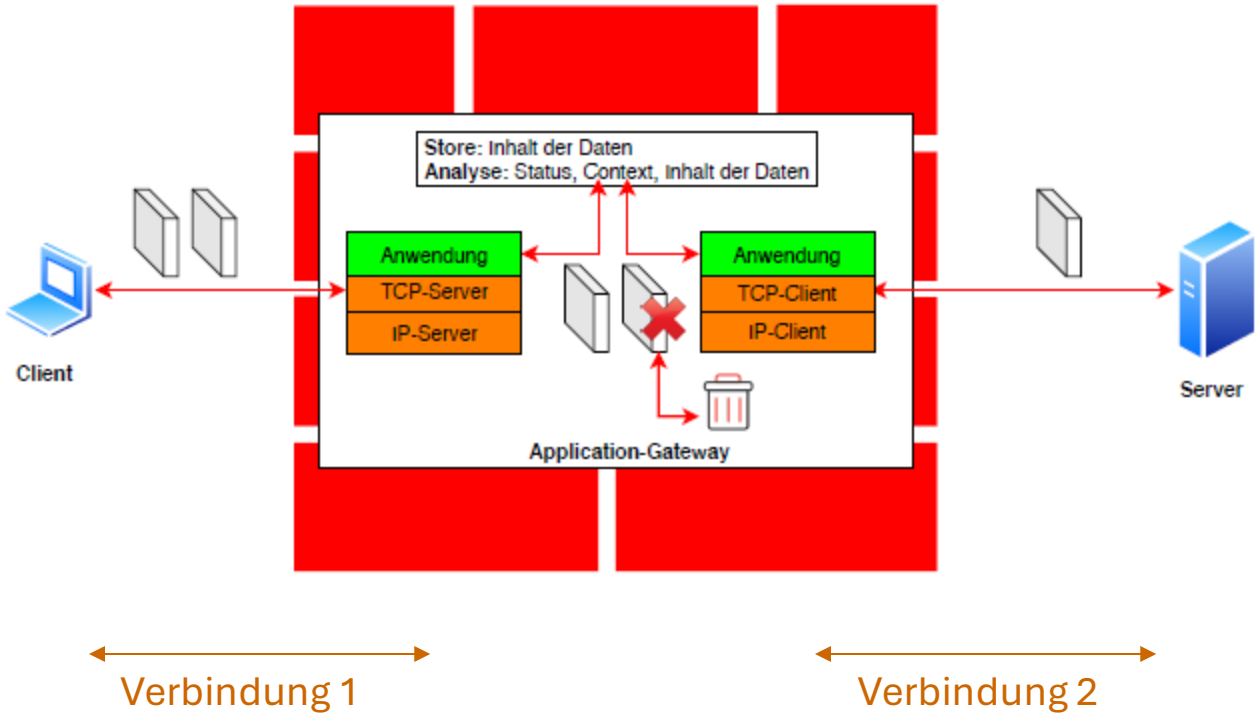
Eine Applikation-Firewall verwendet Proxys, um eine Filterung des Datenverkehrs und eine Zugriffskontrolle durchzuführen.



- Tatsächlich gibt es zwei getrennte Verbindungen zwischen den Verbindungsteilnehmern (z.B.: Web-Browser – Web-Server), wobei sich das Gateway am Trennpunkt-Punkt befindet.

Application-Gateway: Funktionsweise

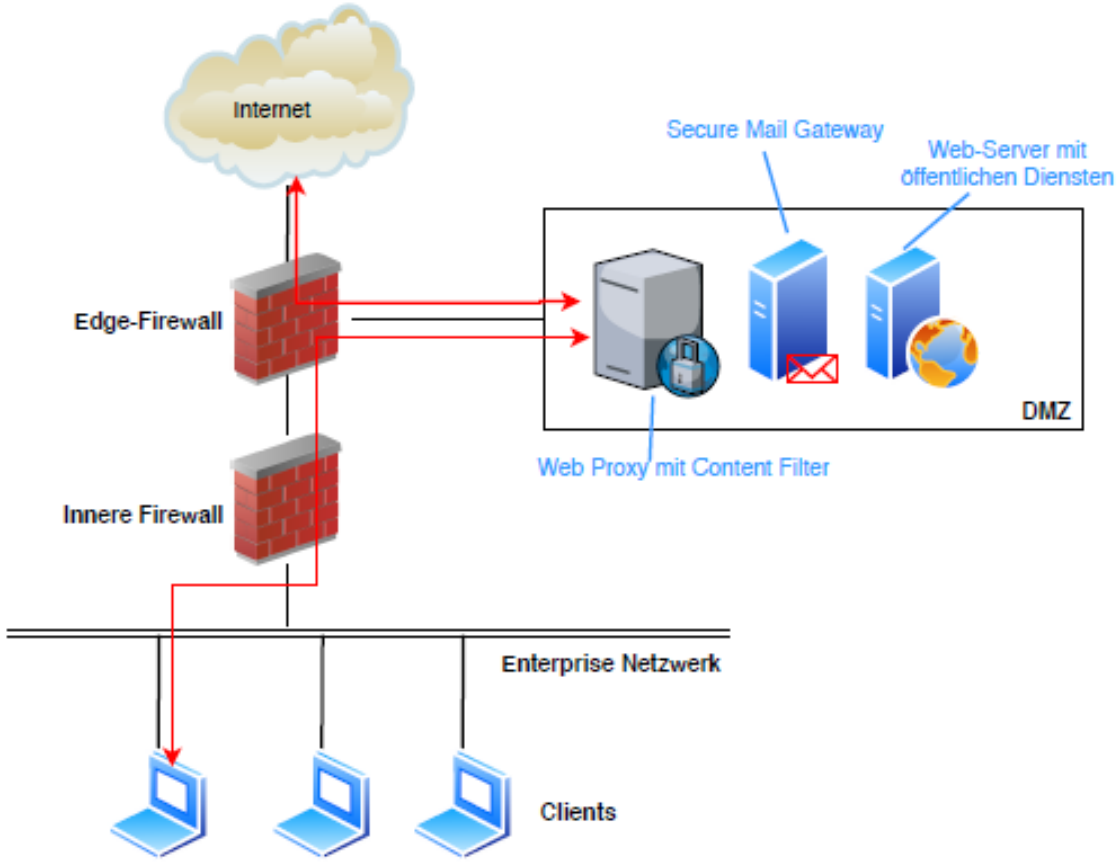
- In den ein- und ausgehenden Pakete werden die Anwendungsnachrichten auf schadhaften oder nicht gewünschten Inhalt geprüft und gefiltert.
- Beispiel Content-Filter:
  - MIME-Anhang einer E-Mail wird in einem Virusscanner gescannt.
- Applikationsgateways ermöglichen auch die Authentifizierung von Benutzern, sodass eine zusätzliche Filterung der Daten auf Anwenderebene möglich ist:
  - Anwendergruppe A darf nur bestimmte Web-Seiten im Internet besuchen.
- Ein Hauptnachteil von Applikations-Gateways ist der zusätzliche Verarbeitungsaufwand für 2 Verbindungen.



Applikations-Gateways und DMZ

Eine DMZ (Demilitarized Zone) ist ein Subnetzwerk, das als Puffzone zwischen einem vertrauenswürdigen internen Netzwerk (z. B. dem privaten LAN einer Organisation) und einem nicht vertrauenswürdigen externen Netzwerk (z. B. dem Internet) fungiert.

- Typischerweise ist die DMZ ist durch eine Edge-Firewall vom Internet und durch eine innere Firewall vom Unternehmensnetzwerk getrennt.
- Durch die Edge-Firewall kann geregelt werden, dass der Datenverkehr von außen nach innen und von innen nach außen nur über die DMZ erlaubt ist.
- Dadurch erfolgt eine Entkopplung vom internen und externen Netzwerk.
- Applikationsgateways in der DMZ analysieren den Datenverkehr und filtern diesen entsprechend vordefinierter Policies.

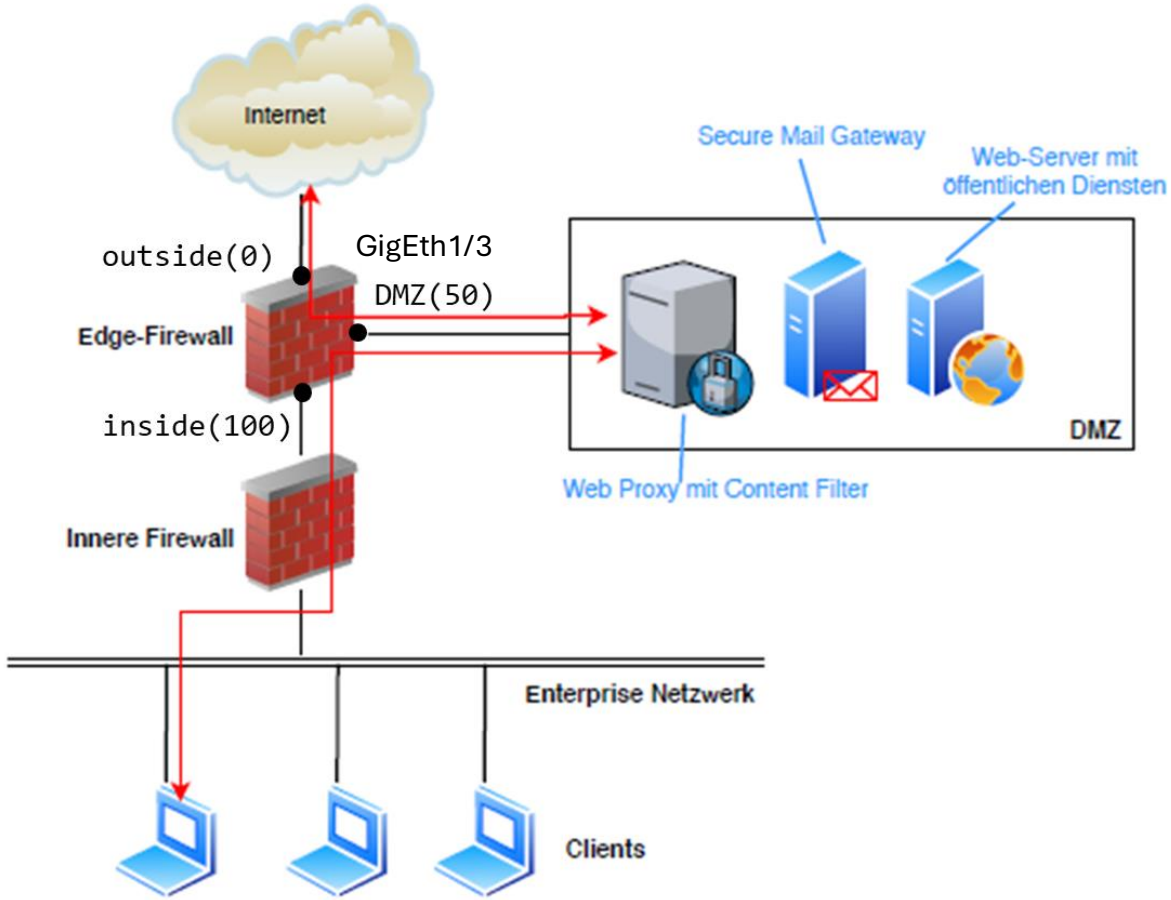


- In einer DMZ werden zusätzlich die Server platziert, die öffentliche Dienste des Unternehmens nach außen anbieten.

Konfiguration einer ASA-Edge-Firewall mit einer DMZ

- Das DMZ-Subnetz wird an ein separates Interface der Edge-Firewall angeschlossen.
- Dieses Interface erhält z.B. den Namen DMZ und einen Sicherheitslevel 50.
- Beispiel: Interface sei GigabitEthernet 1/3

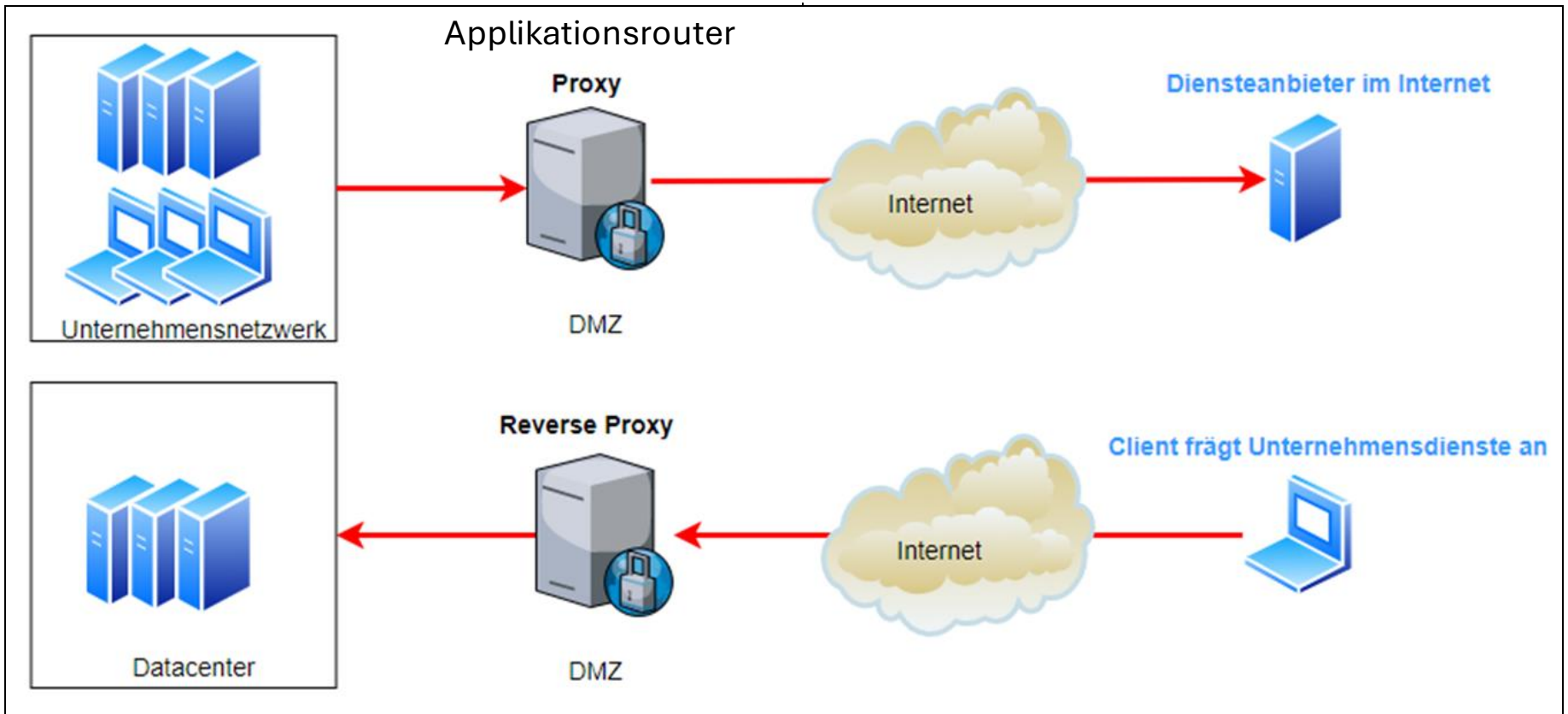
```
FW1(config)#interface GigEth1/3
FW1(config-if)#nameif DMZ
FW1(config-if) #security-level 50
```



Proxy-Server und Reverse-Proxy-Server

- Applikation-Gateways werden oftmals auch als **Proxy-Server** bezeichnet, da Sie stellvertretend für den anfragenden Client, eine Verbindung von „Innen“ nach „Außen“ aufbauen (siehe vorne)

- **Reverse-Proxy-Server** führen stellvertretend die Verbindung von „Außen“ nach „Innen“ durch und ermöglichen somit den kontrollierten Zugriff auf interne Unternehmens-Applikationsdienste.
- Proxys und Reverse-Proxys werden auch als Applikations-router bezeichnet werden.

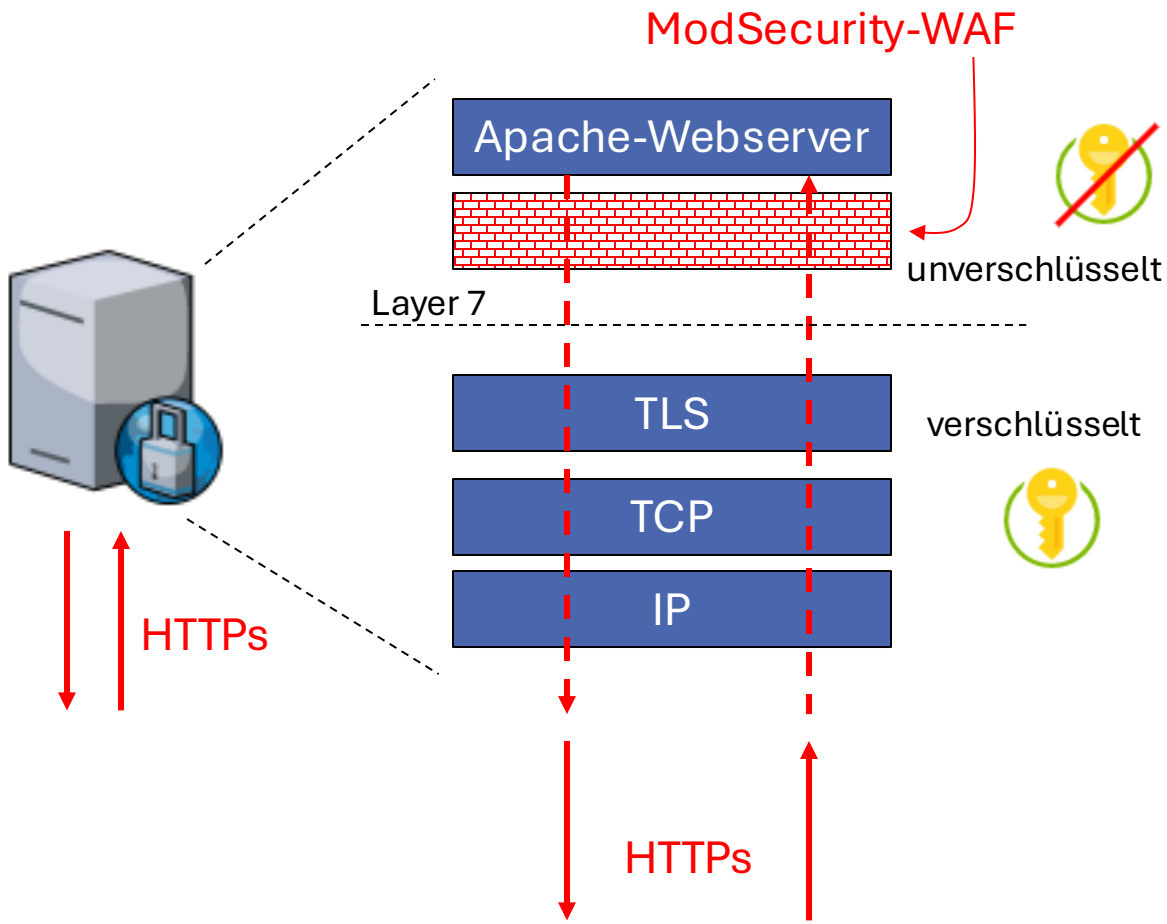


Inline Application Layer Firewalls

- Application-Firewalls analysieren den Datenverkehr typischerweise inline, d.h. die Anwendungspakete müssen das Regelwerk der FW erfolgreich passieren, um den gewünschten Empfänger zu erreichen.
- Beispiel:
  - Web Application Firewall (WAF) untersuchen den HTTP-Header und den HTTP-Body von HTTP-Nachrichten auf Malware und blockieren diese ggf. bevor dieser den Webserver erreicht.
- Damit sind Sie eng verwandt mit
  - Network IPS-Systemen: Untersuchen die ankommenden IP-Pakete inklusive Nachrichten-Payload auf Malware inline.

ModSecurity – Web Application Firewall

- ModSecurity ist eine Open Source Web Application Firewall (WAF).
- Analog zu klassischen Firewall-Systemen basiert ModSecurity auf Regeln.
- ModSecurity läuft direkt auf
  - Apache Web Server
  - NGINX Proxy-Server und Web-Server
  - IIS von Microsoft
- Die OWASP Foundation (Open Web Application Security Project®) ist eine Organisation, deren Ziel es ist die Sicherheit von Webanwendungen zu steigern.
- Die OWASP stellt einen Kernregelsatz für ModSecurity kostenlos bereit.
  https://owasp.org/www-project-modsecurity-core-rule-set/



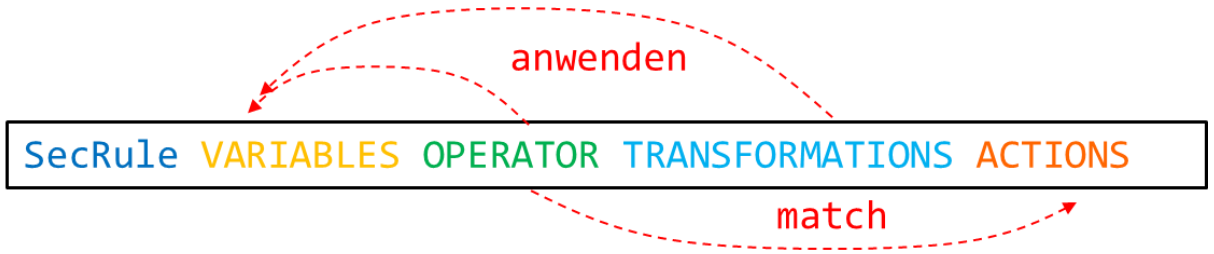
ModSecurity: Rules

- ModSecurity verwendet eine mächtige Regelsprache, um den Webverkehr zu analysieren und ggf. zu blockieren.
- Die Regeln werden in einer zentralen Konfigurationsdatei (modsecurity.conf) gespeichert.
- `SecRule:` Schlüsselwort erstellt eine Regel
- `VARIABLES:` Enthält den Teil einer HTTP-Nachricht, der überprüft werden soll ("where to look")

  `REQUEST_BODY`: Enthält den Body der Nachricht

  `REQUEST_URI`: Enthält die URI der Nachricht

  ...



- `TRANSFORMATIONS`: Normalisiert die Variable bevor der Operator auf sie angewendet wird (z.B.: Entfernung von Steuerzeichen, Leerzeichen, Groß-/Kleinschreibung, ...).
- `OPERATOR:` Gibt an, wie eine (transformierte) Variable analysiert werden soll ("how to look")
  - `"@streq /cgi/malware.php"`: Variable ist gleich mit /cgi/malware.php .
  - `"@contains <script>"`: Variable enthält String-Wert.
  - `"@rx \.php$"`: Regular Expression überprüft ob Variable mit .php (=PHP-Skript) endet.
  - @detectXSS, @detectSQLi: Erkennen von bestimmten Angriffstypen

    ...
- `ACTIONS`: Legt fest, was bei einer Regelübereinstimmung geschehen soll.

Beispiel: ModSecurity-Rules für eine HTTP-URL

- `VARIABLES = REQUEST_URI`: Regel überprüft für jede HTTP-Anforderung die übermittelte URI.
- TRANSFORMATIONS: URI-Wert wird in Kleinbuchstaben umgewandelt und anschließend werden Leerzeichen ("+", "%20") entfernt.
- OPERATOR @contains: Nach der Transformation wird überprüft, ob der transformierte URI-String die Zeichenkette `<script>` enthält (Reflected XSS-Angriff).
- In ModSecurity-Regeln verwenden Sie " " (doppelte Anführungszeichen), um Zeichenfolgen einzuschließen, die Leerzeichen oder Sonderzeichen enthalten.

- ACTION:
  - Im Trefferfall wird die Nachricht verworfen (`deny`) und eine Nachricht (`msg: 'XSS attack detected'`) in das audit.log-File (`log`) geschrieben.
  - Bei Verwendung von Parameter die Leerzeichen oder oder ein Komma enthalten, werden diese in einfache Anführungszeichen gesetzt.
  - Die Regel erhält die eindeutige ID (`id:100000`).

Beispiel: Überprüfen jeder eingehenden HTTP-Request-Nachricht auf XSS-Inhalt in der übermittelten HTTP-URI

```
SecRule REQUEST_URI "@contains <script>"  \
                    "t:lowercase, t:removeWhitespace, \
                     id:100000,phase:1,deny,log,msg:'XSS attack detected'"
```

ModSecurity: Audit.log

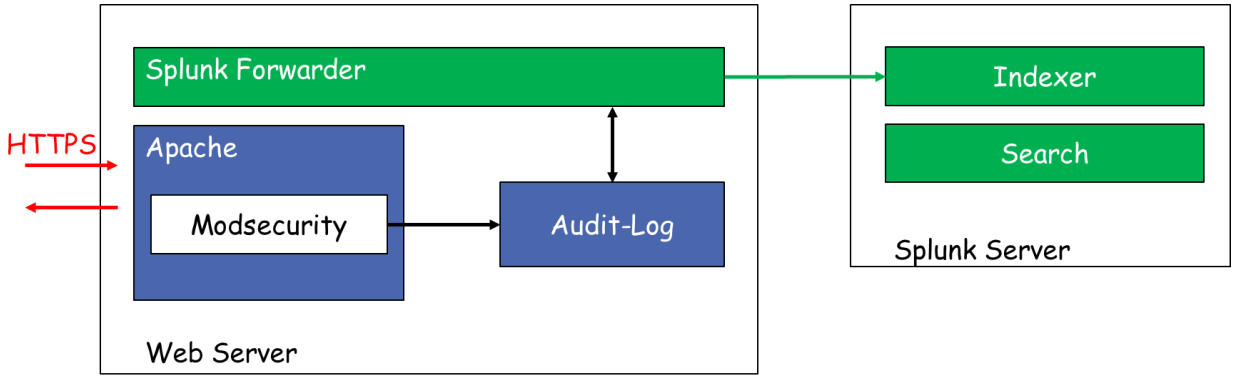
- ModSecurity enthält ein Tool namens mlogc (kurz für ModSecurity Log Collector), mit dem Audit-Protokolle in unmittelbar an einen Remote-Protokollierungsserver übertragen werden können.
- Hierzu muss im Konfigurationsfile des Tools (typisch: `/etc/modsecurity/mlogc.conf/mlogc.conf`), die Adresse des Remote-Server-Protokollierungsserver (z.B.: SIEM-System Splunk) inklusive der benötigten Anmeldeinformation eingetragen werden

```
# Remote logging server details.
ConsoleURI "https://www.eample.de:8888/auditLogReceiver"
Username "USERNAME"
Password "PASSWORD"
```

Ein SIEM-Server (Security Information and Event Management-Server) ist eine zentrale Datenplattform zum Sammeln, Analysieren und Verwalten sicherheitsrelevanter Daten aus verschiedenen Quellen innerhalb einer Organisation.

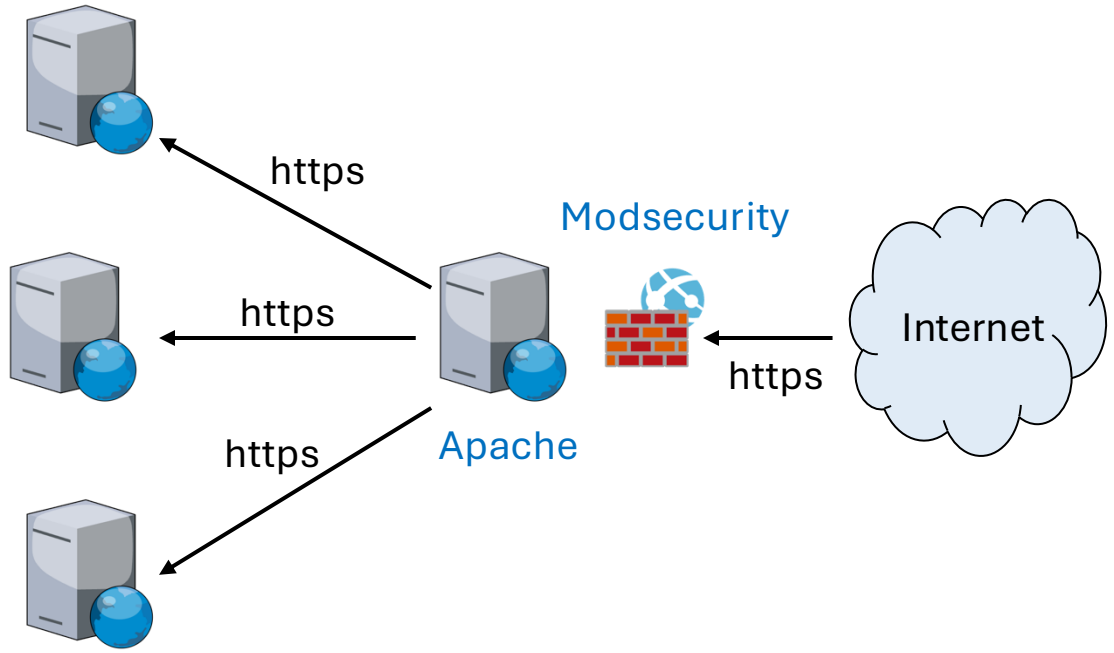
- Eine andere Möglichkeit zum Transfer der Log-Dateien auf einen SIEM-Server besteht darin, lokale Agents eines SIEM-Server auf dem Web-Server mit ModSecurity zu installieren.
- Beispiel: Splunk Forwarder

  Splunk Forwarder liest die von ModSecurity erstellten Log-Files ein und transferiert diese zum Splunk-Server.



Einsatzszenario: Apache & ModeSecurity als Reverse-Proxy

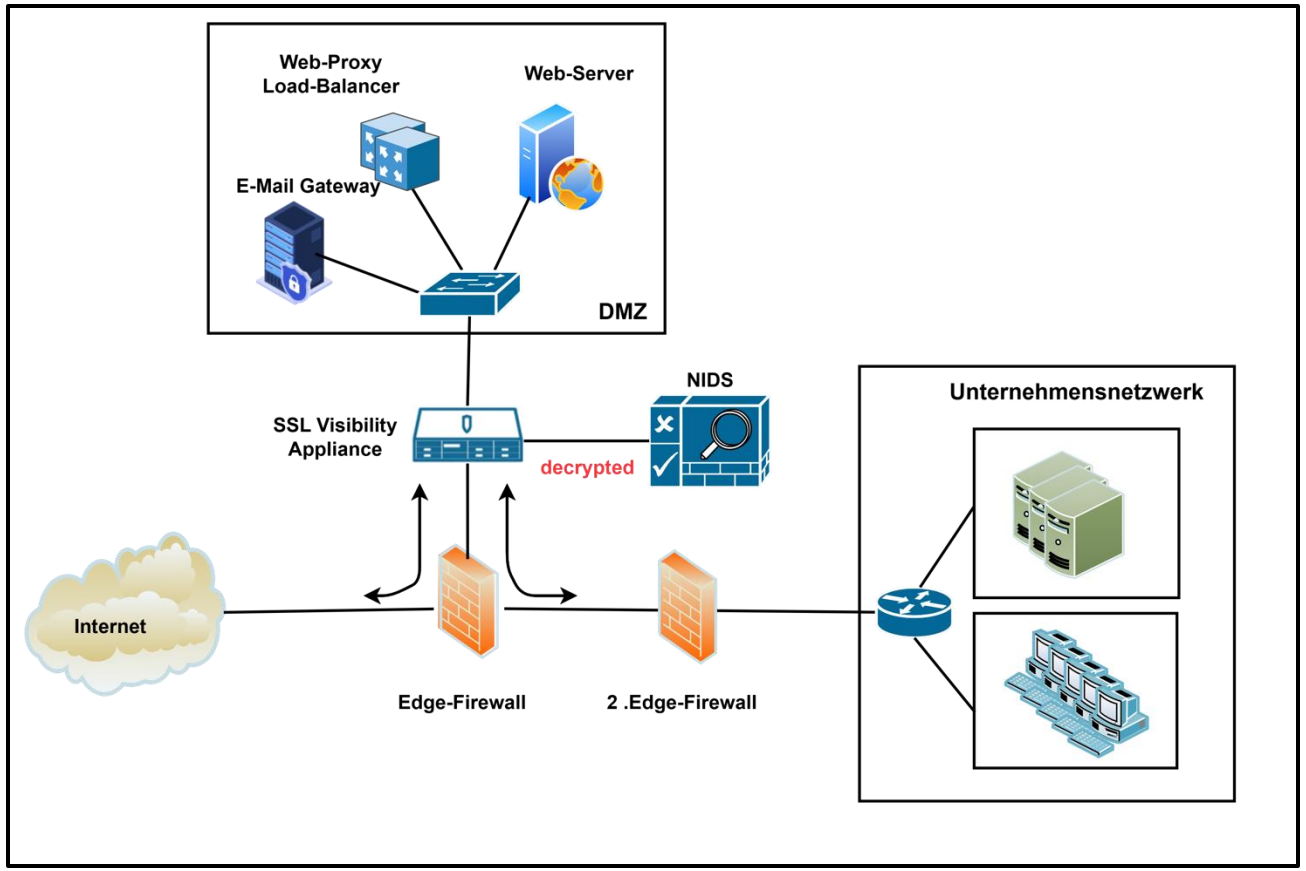
- Apache kann in Kombination mit Modsecurity als eine zentrale Web Application Firewall in der DMZ installiert werden.
  - Modsecurity analysiert den HTTP-Verkehr auf schädlichen Content und verwirft diesen ggf.
  - Apache arbeitet als Reverse-Proxy, der den eingehenden und unschädlichen Verkehr an die Webserver im Unternehmen verteilt.



2.3 Intrusion Detection and Prevention Systeme

NIPS, NIDS, SSL Visibility

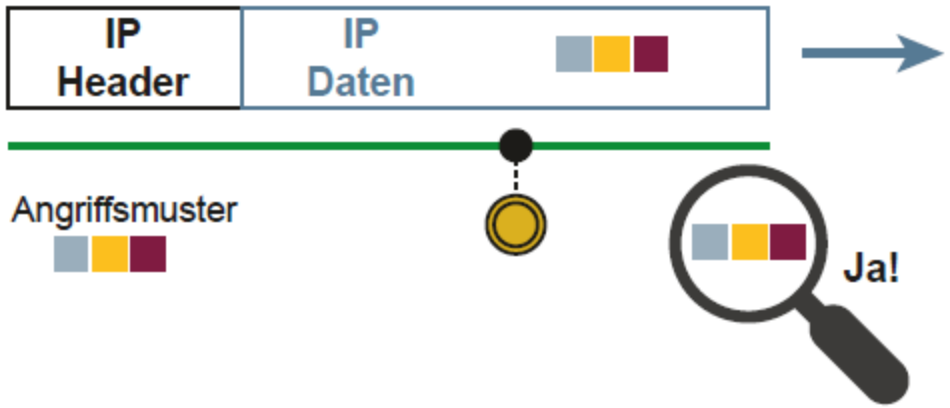
Appliance



Aufgaben von IPS- und IDS-Systemen

- Hacker versuchen über das Netzwerk (Internet, LAN, WLAN, ...) Zugriff auf Systeme und Daten eines Unternehmens zu erlangen.
- Um einen Einbruchsversuch zu erkennen, können sogenannte Intrusion Detection Systeme (IDS) verwendet werden.
- Sind die Systeme auch in der Lage einen erkannten Angriff zu verhindern, werden Sie als Intrusion Prevention Systeme (IPS) bezeichnet.
- IPS- und IDS-Systeme analysieren den kompletten Netzwerkstack eines IP-Paketes auf ein mögliches Angriffsmuster anhand eines vordefinierten Regelwerkes.
- Moderne IPS-/IDS-Systeme verwenden zudem Machine Learning-Verfahren, um Anomalien im Netzwerkverkehr zu erkennen, und diese dann als Angriff zu interpretieren.

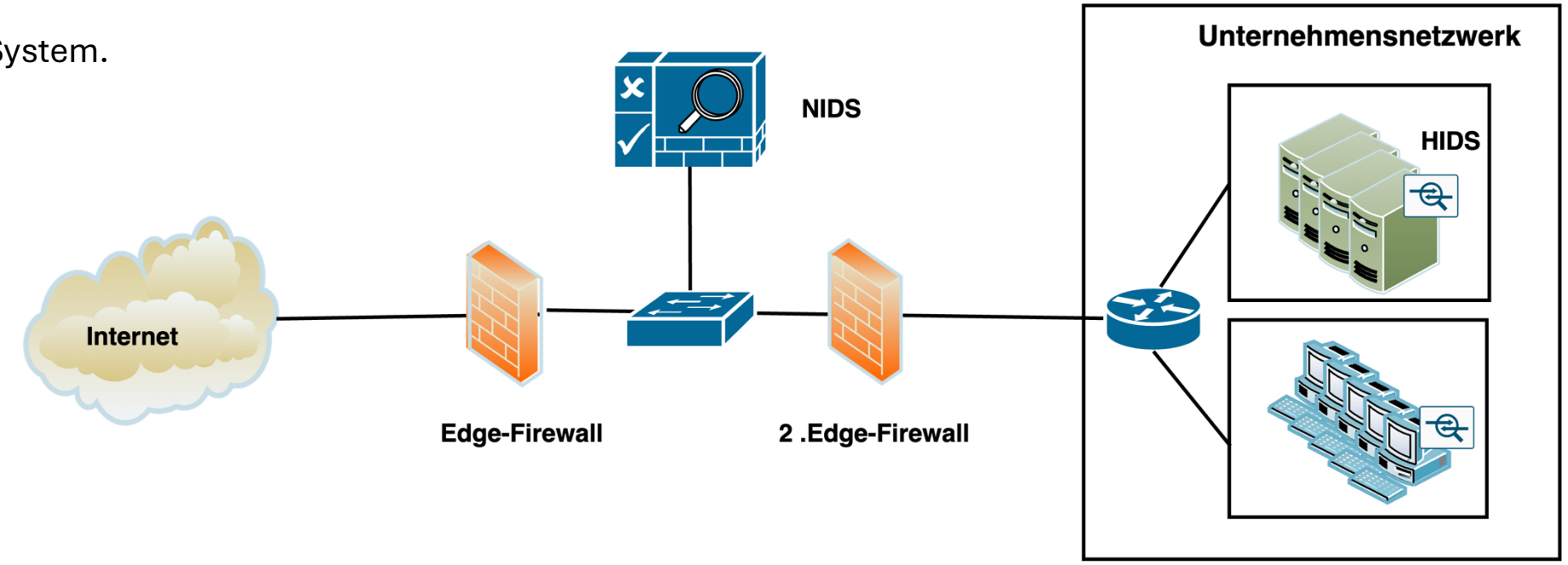
Beispiel: Analyse eines IP-Paketes auf ein mögliches Angriffsmuster im Payload.



Typen von Intrusion Detection/Prevention Systeme (IDS)

- Das Erkennen von Angriffen kann entweder lokal auf einem System oder im Netzwerk erfolgen.
- Hostbasierte Intrusion Detection Systems (HIDS) sind Systeme, die Daten analysieren, welche auf einem einzelnen Host aufgezeichnet werden.
  - HIDS-Systeme lesen die lokalen Audit-Log-Dateien mittels SW-Agenten und transferieren diese zu einem zentralen HIDS-Analyse-System.

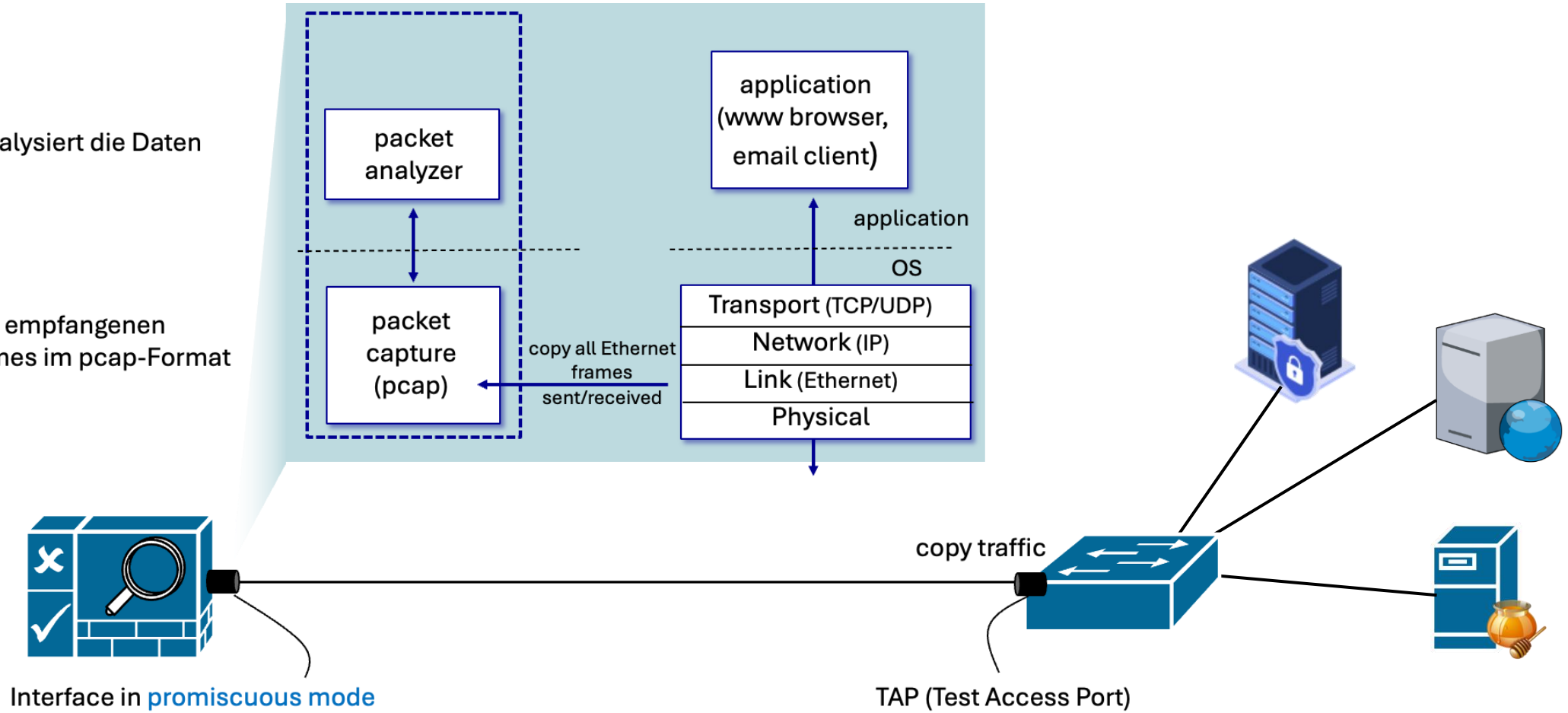
- Netzwerkbasierte Intrusion Detection Systems (NIDS) sind Systeme, die den Netzwerkverkehr an zentralen Knotenpunkten (Übergang zum Internet, Übergang zu Produktionssystemen, ...) beobachten und die dabei gefundenen Pakete auf Angriffsmuster untersuchen.



Packet Capturing

Filtert und analysiert die Daten im pcap-File.

Speichert die empfangenen Ethernet Frames im pcap-Format

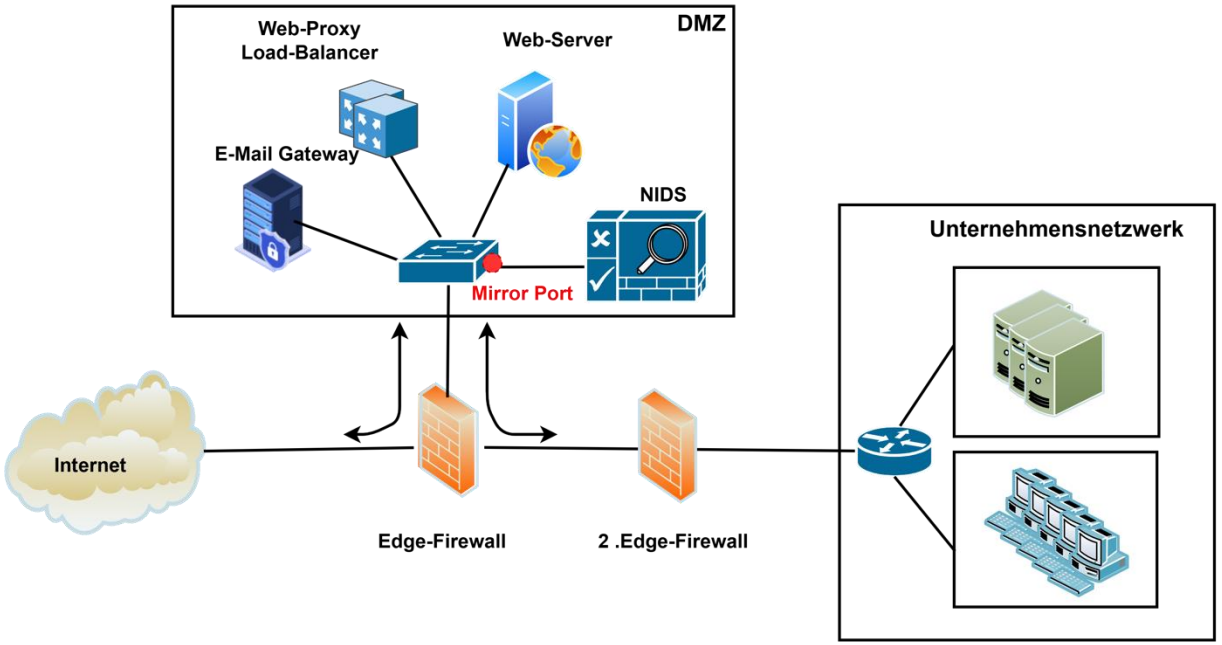


pcap: packet capture format

promiscuous mode: accept each ethernet frame

NIDS: Perimeter Netzwerkverkehr

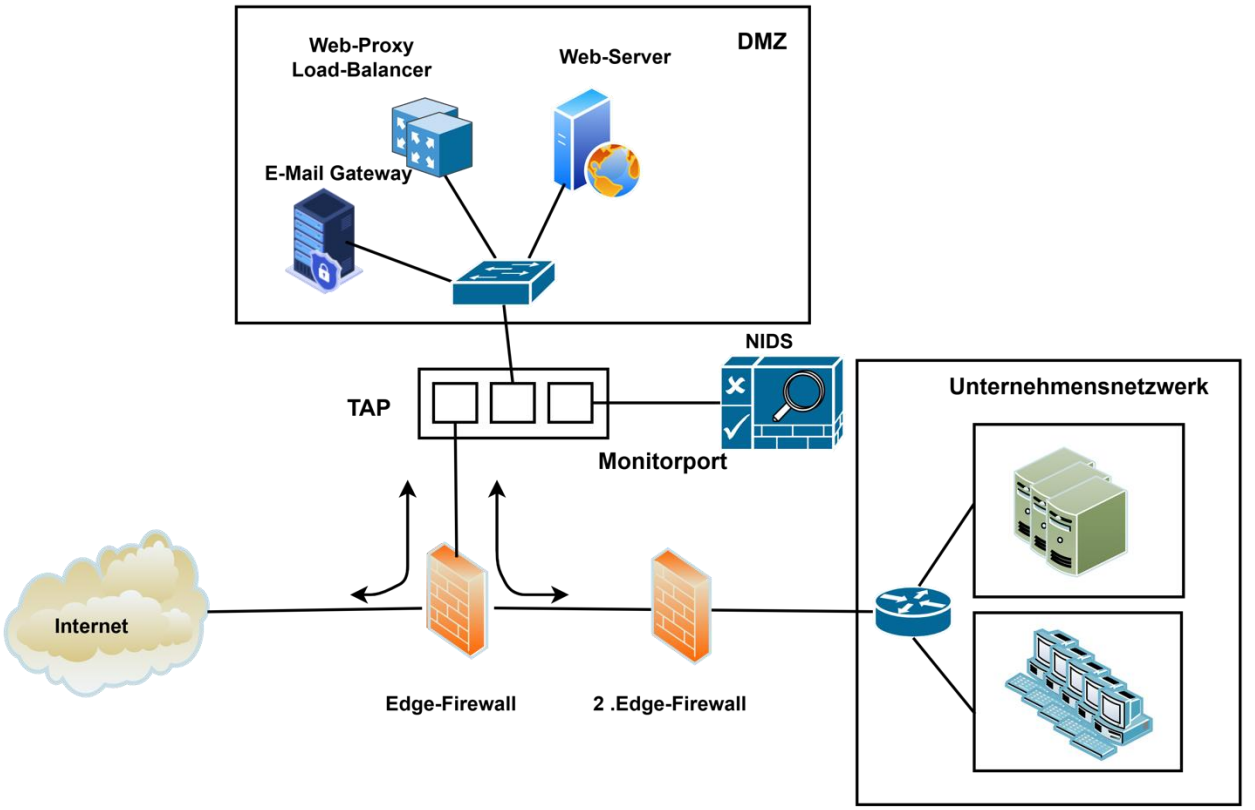
- Um den Perimeter ihres Unternehmensnetzwerkes zu schützen, überwachen NIDS-Systeme den eingehenden und ausgehenden Netzwerkverkehr in der DMZ.
- Dazu können NIDS-Systeme zum Beispiel an den Mirror-Port (SPAN[1]-Port) eines Netzwerk-Switches angeschlossen werden und erhalten so den gespiegelten Datenverkehr, der durch den Switch fließt.
- Ein NIDS-System befindet sich out-of-band und analysiert die Kopien der Originalpakete (promiscuous mode).
- Wichtig: Switch priorisiert den normalen Daten-Verkehr gegenüber Mirror Port-Traffic. Einzelne Packete können je nach Lastsituation nicht gespiegelt werden.



SPAN[1]: Switched Port Analyser

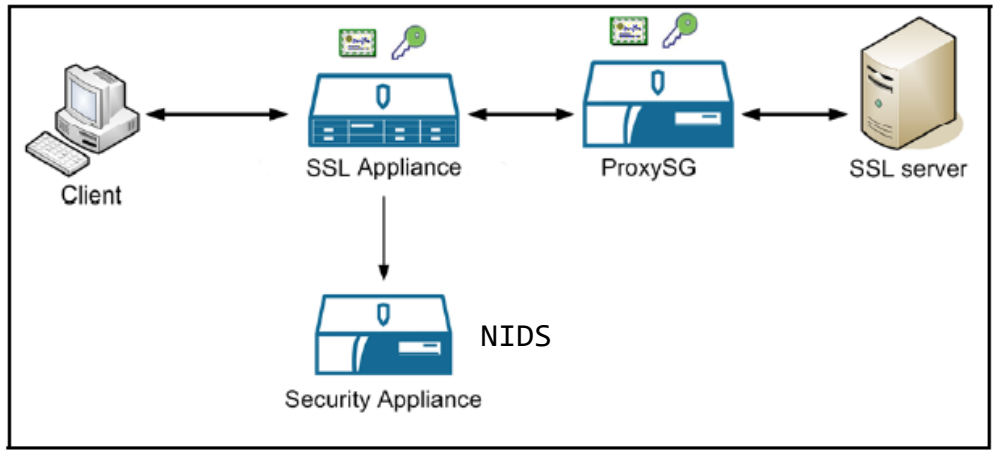
NIDS: Perimeter Netzwerkverkehr und TAPs

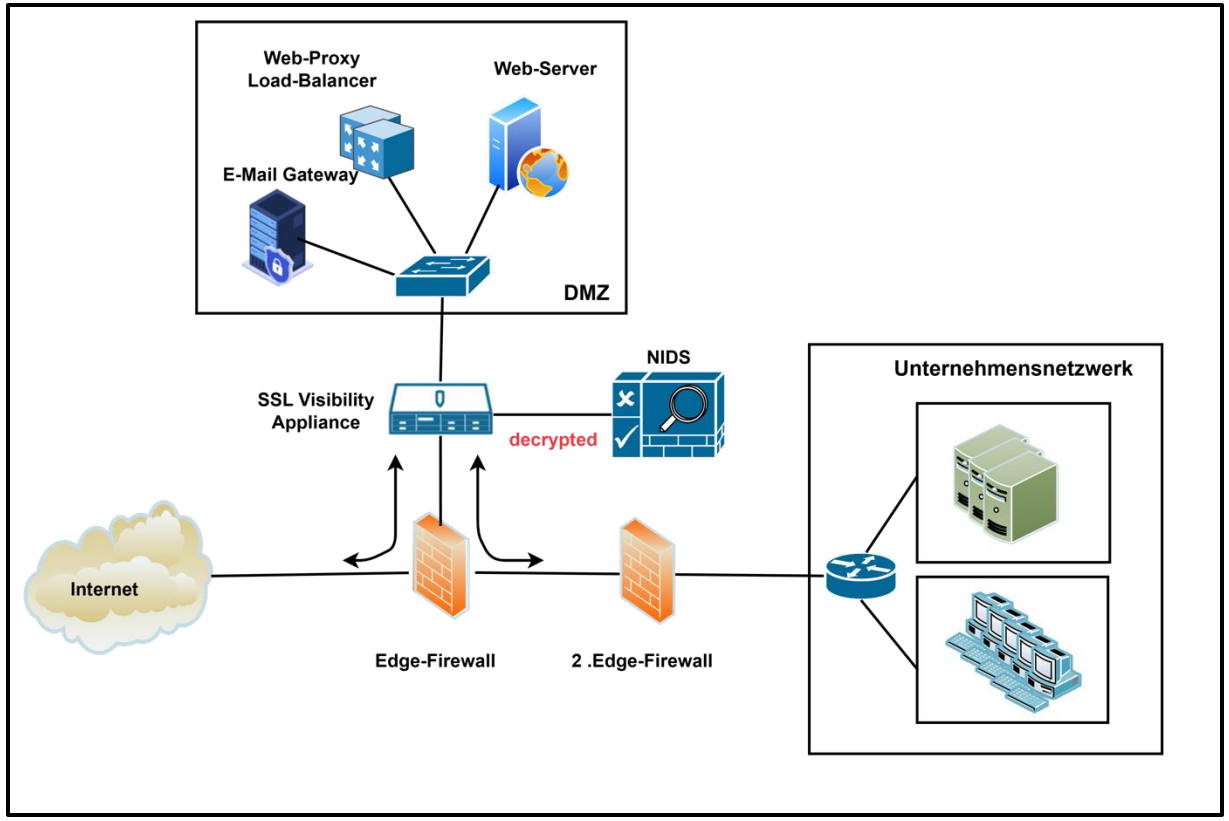
- TAPs (Test Access Points), sind eigenständige Hardware-geräte, die eine 100%-Kopie des gesamten Datenverkehrs erstellen, der zwischen zwei Endpunkten in einem Netzwerk fließt.
- TAPs arbeiten auf dem Layer-1 während SPAN-Ports auf Layer-2
- TAPs spiegeln bidirektionale Datenströme gleichzeitig und sind von Bandbreiten und Auslastung unabhängig, wodurch das Risiko von Paketverlusten vollständig eliminiert, wird.
- Empfehlung: In sicherheitskritischen Umgebungen sollten TAPs eingesetzt werden.



NIDS & SSL Visibility Appliance

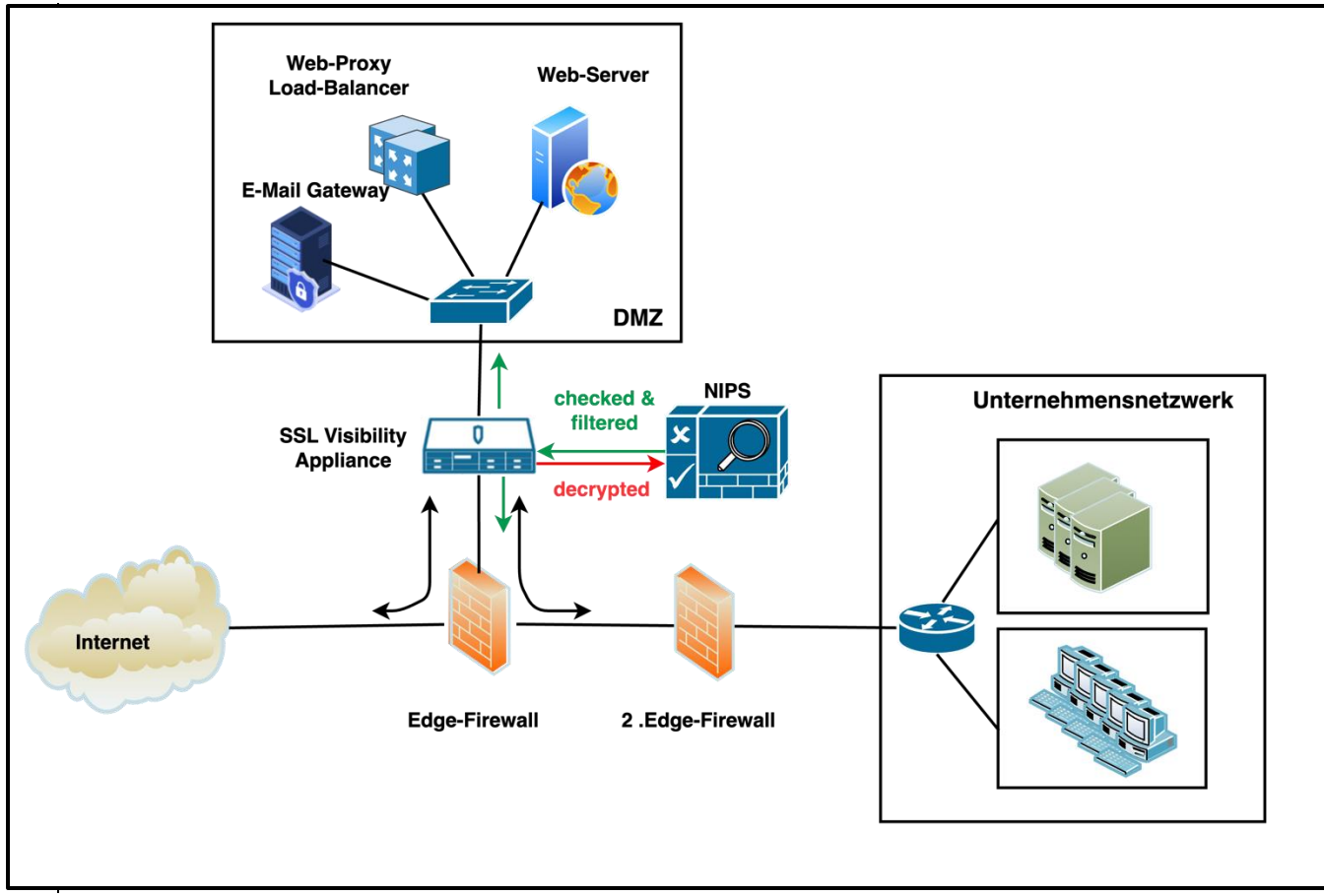
- Um auch verschlüsselten Datenverkehr analysieren zu können, kommen sogenannte SSL Visibility Appliances zum Einsatz.
- Die SSL Visibility Appliance arbeitet als MITM-Gerät.
- Die SSL Visibility Appliance baut eine verschlüsselte Verbindung zum Host im Internet und zum Web-Proxy-Server in der eigenen DMZ auf.
- Der entschlüsselte Datenverkehr wird an ein NIDS-System über einen Mirror-Port zur Analyse weitergeleitet (passiver Modus).
- Das NIDS – System analysiert den Netzwerkverkehr und sendet im Falle einer entdeckten Intrusion, die zugehörigen Alarme an ein SIEM-System.





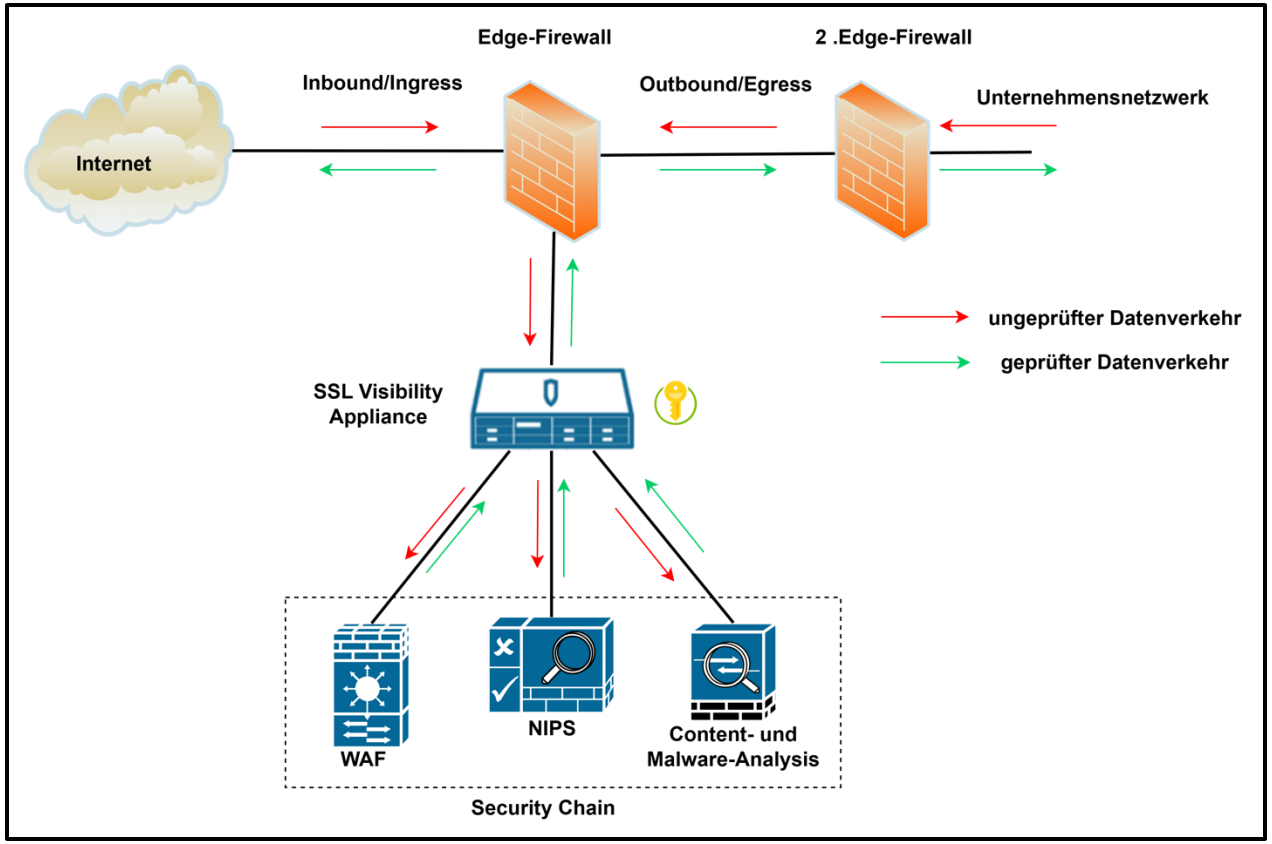
NIPS & SSL Visibility Appliance

- Durch den Anschluss eines Network-IPS-Systems an die SSL Visibility Appliances, kann der vom Internet eingehende Netzwerkverkehr analysiert und ggf. gefiltered werden.
- Das IPS-System analysiert den unverschlüsselten Datenverkehr der SSL Visibility Appliance in Echtzeit und filtert basierend auf einem Regelsatz mögliche schadhafte Datenpakete (aktive Modus oder Inline-Modus).
- Die SSL-Appliance leitet nur die vom IPS-System zurückgelieferten Datenpakete in das jeweilige Netzwerk weiter
  - von innen nach außen
  - von außen nach innen
- Ein NIPS-System befindet sich Inline, d.h. der Datenverkehr muss durch das NIPS-Systemen fließen.



SSL Visibility Appliance & Security Chain

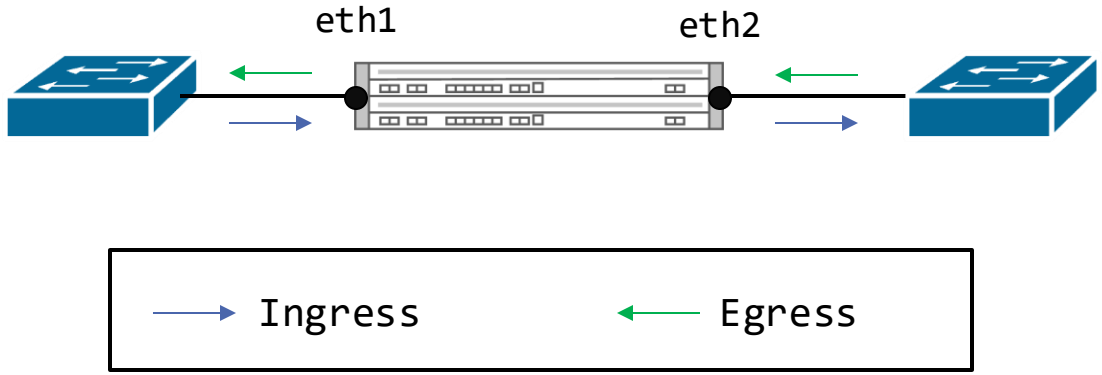
- An die SSL Visibility Appliance können weitere Sicherheits-dienste angeschlossen werden, die dann nacheinander durchlaufen werden. Man spricht dann von einer Security-Chain.
  - Neben einer Web Application Firewall (WAF), die den HTTP-Verkehr auf Layer-7 analysiert, werden Malware-Filter und Content-Filter eingesetzt.
- Ausprägung von Content-Filtern
  - Web-Content-Filter: Filtert bestimmte Web-Seiten und deren Inhalte (Text, Bilder, Dokumente).
  - E-Mail-Content-Filter: Filtert in E-Mails deren textuelle Inhalt und deren Anhänge.



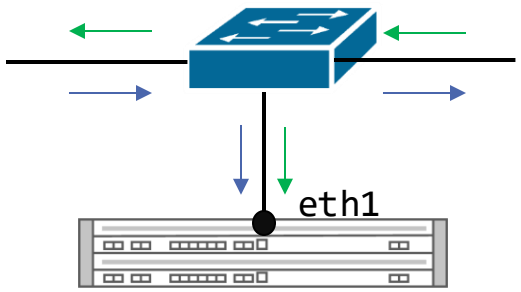
Zusammenfassung: NIDS versus NIPS

- NIPS-Systeme
  - arbeiten im inline-Mode.
  - sehen die Originalpackete und können diesen gemäß ihrem internen Regelwerk durchlassen oder löschen.
  - arbeiten in Echtzeit
  - erzeugen in der Gesamtverarbeitung des Packetes einen Processing Overhead: erst analysieren und dann weiterleiten
- NIDS-Systeme
  - arbeiteten im out-of-band-Mode.
  - sehen Kopie der Originalpackete und können diese analysieren und auf Basis ihres internen Regelwerkes alarmieren.
  - Angriffe werden nur gemeldet.

NIPS-Systeme



NIDS-Systeme



NIDS: Signaturbasierende und statistische Analysen von Netzwerk-Paketen

- Signaturbasierendes Verfahren: Bei der Signatur-Erkennung wird der Inhalt der IP-Packete gelesen und mit einem Regelwerk verglichen.
  - Specification-Rules: Abgleich mit der Protokoll-Spezifikation

    ⟹ Entspricht der Aufbau, Inhalt und Größe des Nachrichtenheaders (Layer-7) den RFC-Empfehlungen oder Herstellerempfehlungen?

    ⟹ Wurden die Protokollheader (Layer-2, Layer-3, Layer-4) gemäß den RFC-Empfehlungen umgesetzt?
  - MissUse-Rules: Abgleich mit Angriffssignaturen

    ⟹ Ist der Nachrichteninhalt (Layer-7) für Malware typisch (SQL-Injection, XSS, ...).

- Anomalie basierende Verfahren: Basieren auf Machine Learning-Analysen, die den eingehenden Datenverkehr mit einem Referenz-Verkehr (Baselining) vergleichen. Bei Abweichung wird auf einen Angriff geschlossen.

  ⟹ Berücksichtigung der Base-Rate-Fallacy

- Während signaturbasierende Verfahren nur bekannte Angriffe erkennen können, ermöglichen statistische Verfahren auch das Erkennen bzw. Erlernen von neuen Angriffsmustern.

Architektur von IDS-Systemen

- Die Abbildung zeigt die Verarbeitung der Netzwerk- oder Audit-Log-Daten in einem NIDS- oder HIDS-System.
- Für ML-Verfahren und zur Effizienzsteigerung filtert man die Daten vor deren Bewertung.

Output Module: Anzeige eines Alerts in Dashbord, E-Mail-Versand, SMS-Versand,....

Anomalie Detection Module: Anwendung des Regelwerkes auf den Payload der aufbereiteten Pakete.

Match → Alert, Drop, Log

Präprozessor: Normalisierung der gesammelten Daten, Analyse der Protokollheader mit Specification-Rules. Weitergabe sicherheitsrelevanter Daten.

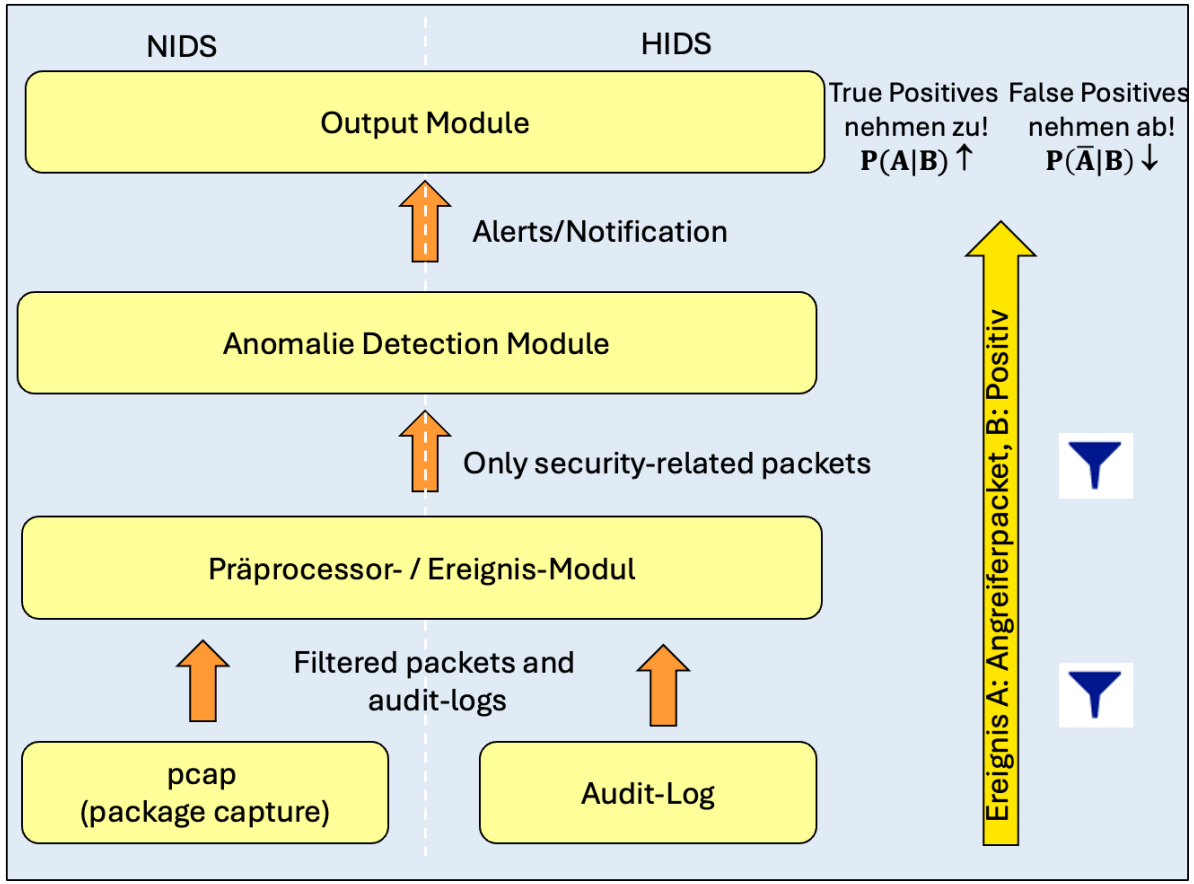
Datenerfassung: Sammeln und filtern sicherheitsrelevanter Daten.

A: Angriff, $\overline{A}$: Kein Angriff

B: positiv bewertet (=Angriff), $\overline{B}$: negativ bewertet

P(A|B): True Positive (richtig erkannter Angriff)

$P(\overline{A}|B)$: False Positive (falsch erkannter Angriff)



Prävalenzfehler (Base-Rate-Fallacy)

Als Prävalenzfehler bezeichnet man den Fehler, der entsteht, wenn die Bestimmung der bedingten Wahrscheinlichkeit P($\overline{A}$ |B) einer statistischen Variable $\overline{A}$ unter einer Bedingung B ohne Berücksichtigung der Prävalenz (Häufigkeit) P($\overline{A}$) von $\overline{A}$ in den Daten vorgenommen wird.

- Beispiel: Die False Positiv Rate FP für ein IPS-System ist gegeben durch die Häufigkeit der Anwenderpakete P($\overline{A}$) multipliziert mit der Wahrscheinlichkeit, dass das IPS-System das Anwenderpaket als Angriff erkennt:

$$FP = P(\overline{A}\,|B) \sim P(B|\overline{A}) \cdot P(\overline{A})$$

Fehlerquote (Eigenschaft IPS)

Base Rate $\overline{A}$ (Eigenschaft des Netzwerkverkehrs)

- Am Beispiel erkennt man, das bei einer hohen Anzahl von normalen Paketen $P(\overline{A})$ im aufgezeichneten Datenstrom, selbst bei einer guten Präzession $P(B|\overline{A})$ des IPS-Systems eine hohe False Positiv Rate FP entsteht.
- Für eine hohe True Positive Rate TP ist eine hohe Präzession $P(B|A)$ des Systems notwendig.

$$TP = P(B|A) \sim P(B|A) * P(A)$$

Trefferquote (Eigenschaft IPS)

Base Rate A (Eigenschaft des Netzwerkverkehrs)

$$P(B|\overline{A}) + P(B|A) = 1$$

Ereignis A: Angreiferpaket;

Ereignis $\overline{A}$: Anwenderpaket;

Ereignis B: Paket wird als Angriff bewertet (positiv);

Ereignis $\overline{B}$: Paket wird als Normal bewertet (negativ);

Snort: Ein IDS/IPS-System

- Snort gehört der Firma CISCO
- Bei Snort handelt es sich um ein OpenSource NIDS/NIPS – System, das weite Verbreitung gefunden hat und mittlerweile Bestandteil vieler kommerzieller Systeme ist (z.B.: CISCO ASA Firewalls).

```
https://www.snort.org/
```

- Zentrale Snort-Einstellungen werden in der folgenden Datei konfiguriert:

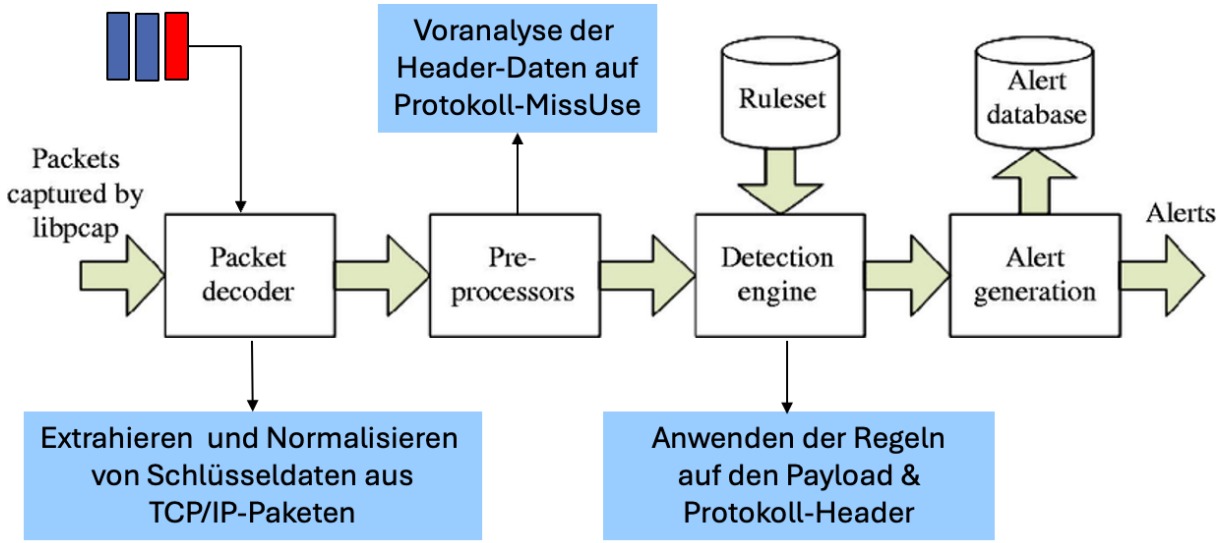
```
/etc/snort/snort.conf
```

- Snort bringt eine große Anzahl an vordefinierten Regeln zur Erkennung von Cyber-Attacken mit und ermöglicht es dem Anwender eigene Regeln zu definieren.

- Bezüglich des enthaltenen Regelwerks gibt es 2 Abonnentenmodelle
  (1) Snort Subscriber Rule Set: Regelwerk wird von der Firma CISCO weiterentwickelt, getestet und zeitnah ihren Abonnenten zur Verfügung gestellt.
  CISCO stellt seine Regeln mit einem Zeitversatz von 30 Tagen der Community zur Verfügung.
  (2) Community Rule Set: Das Regelwerk wird von der Snort-Community weiterentwickelt.

Snort

- **1 Paketerfassung**
  - Snort liest Datenpakete über seine Netzwerkschnittstelle.
  - Diese Pakete werden im pcap-Format an den Packet Decoder weitergeleitet.
- **2 Extrahieren und Normalisieren (Packet Decoder)**
  - Der Decoder überprüft das Paket auf Integrität und Korrektheit.
  - Die Header-Daten (z. B. Ethernet-Header, IP-Header, TCP/UDP-Header) im Paket und die Nutzdaten werden extrahiert und normalisiert.
  - Fehlerhafte oder unvollständige Pakete werden verworfen.
- **3 Vorverarbeitung (Preprocessors)**
  - Preprocessoren sind SW-Module, die Pakete vorverarbeiten und modifizieren können, um Angriffe wie Fragmentierungsangriffe oder Evasion-Techniken zu erkennen.
  - Beispiele für Preprocessors:



Frag3: Rekonstruiert fragmentierte IP-Pakete.

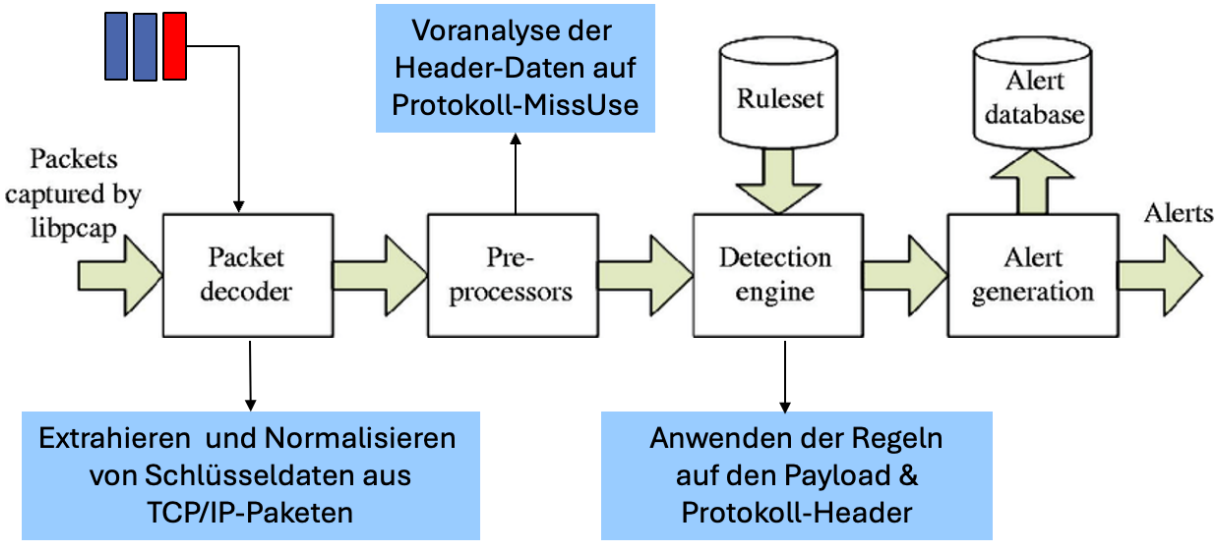
Stream5: Führt TCP-Stream-Reassemblierung durch.

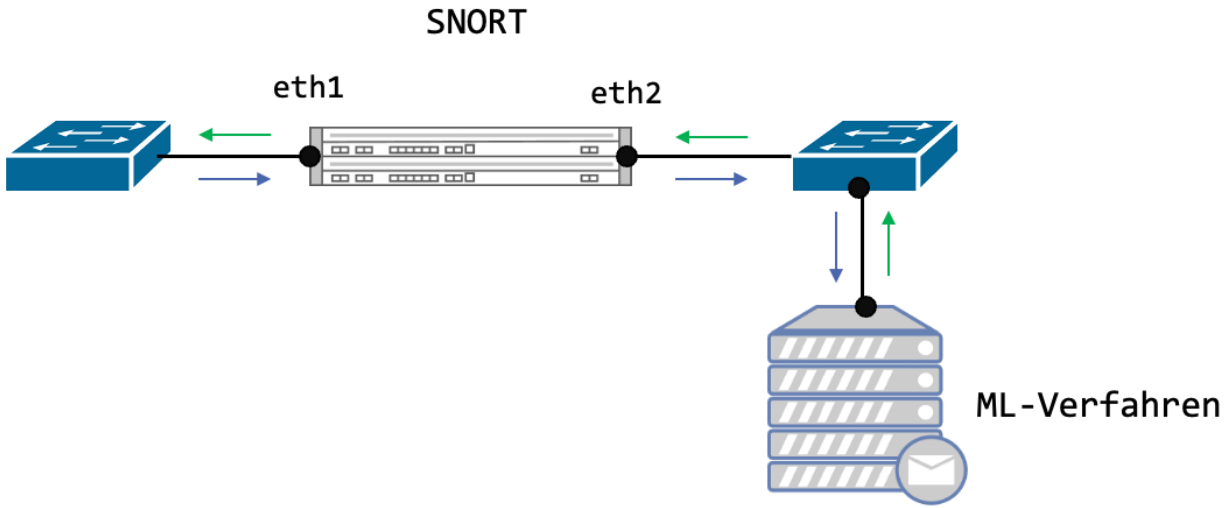
HTTP Inspect: Decoded und analysiert HTTP-Datenverkehr, um ungewöhnliches Verhalten oder bekannte Angriffsvektoren zu erkennen.

- Die Präprozessoren können Alarmmeldungen auslösen, wenn Auffälligkeiten erkannt werden.

Snort

- **4 Regelabgleich (Detection Engine)**
  - Die Detection Engine prüft das Paket gegen die Snort-Regeln.
  - Snort-Regeln bestehen aus zwei Teilen:
    1. Header: Enthält grundlegende Informationen wie Protokoll, IP-Adressen, Ports und Richtung.
    2. Optionen: Definiert detaillierte Bedingungen, z. B. Inhalte, Muster, Flags und Aktionen.
- **5 Alarmierung (Alert Engine)**
  - Sende Alarme oder Logs an ein zentrales Monitoring-System: zentraler Syslog-Server oder zentrales SIEM-System.
- **Optional:**
  - Weiterleitung an ein ML-System, um unbekannte Angriffe zu erkennen.

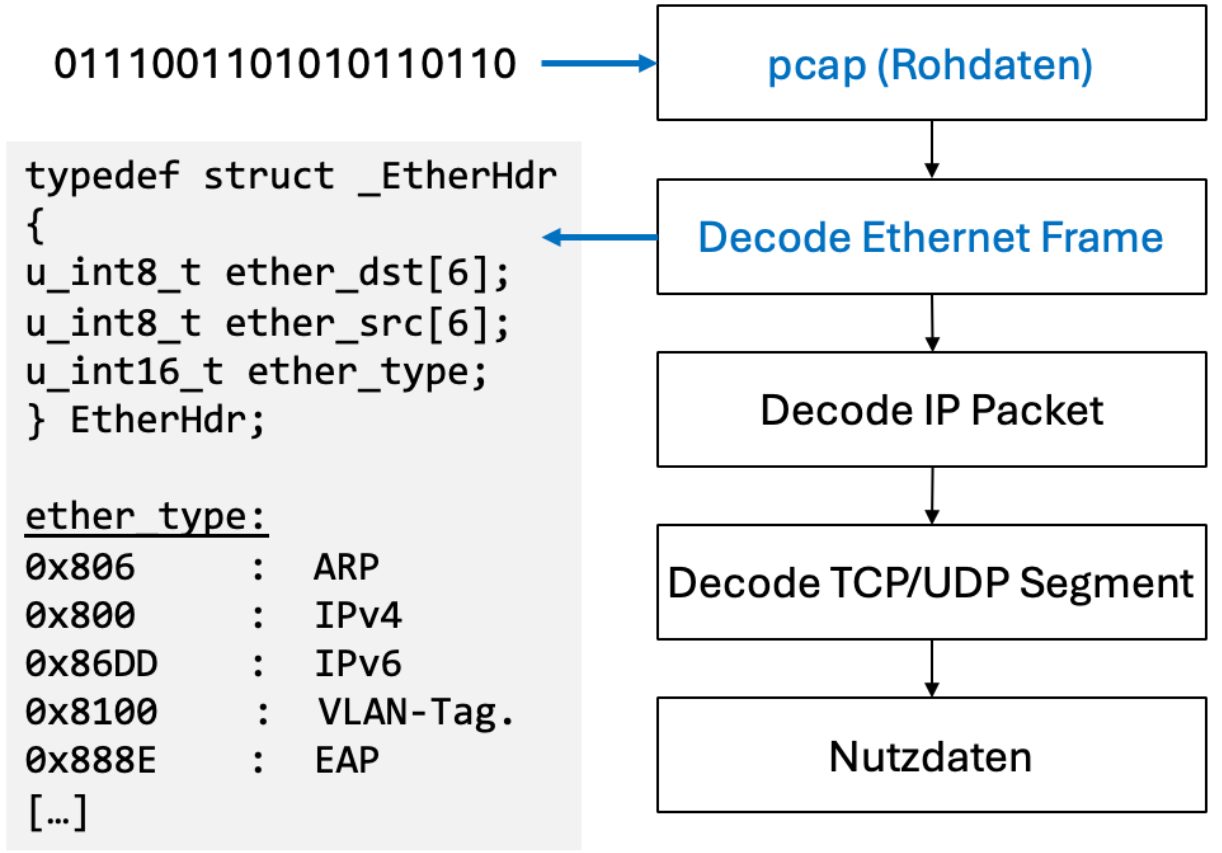




Snort Packet Decoder

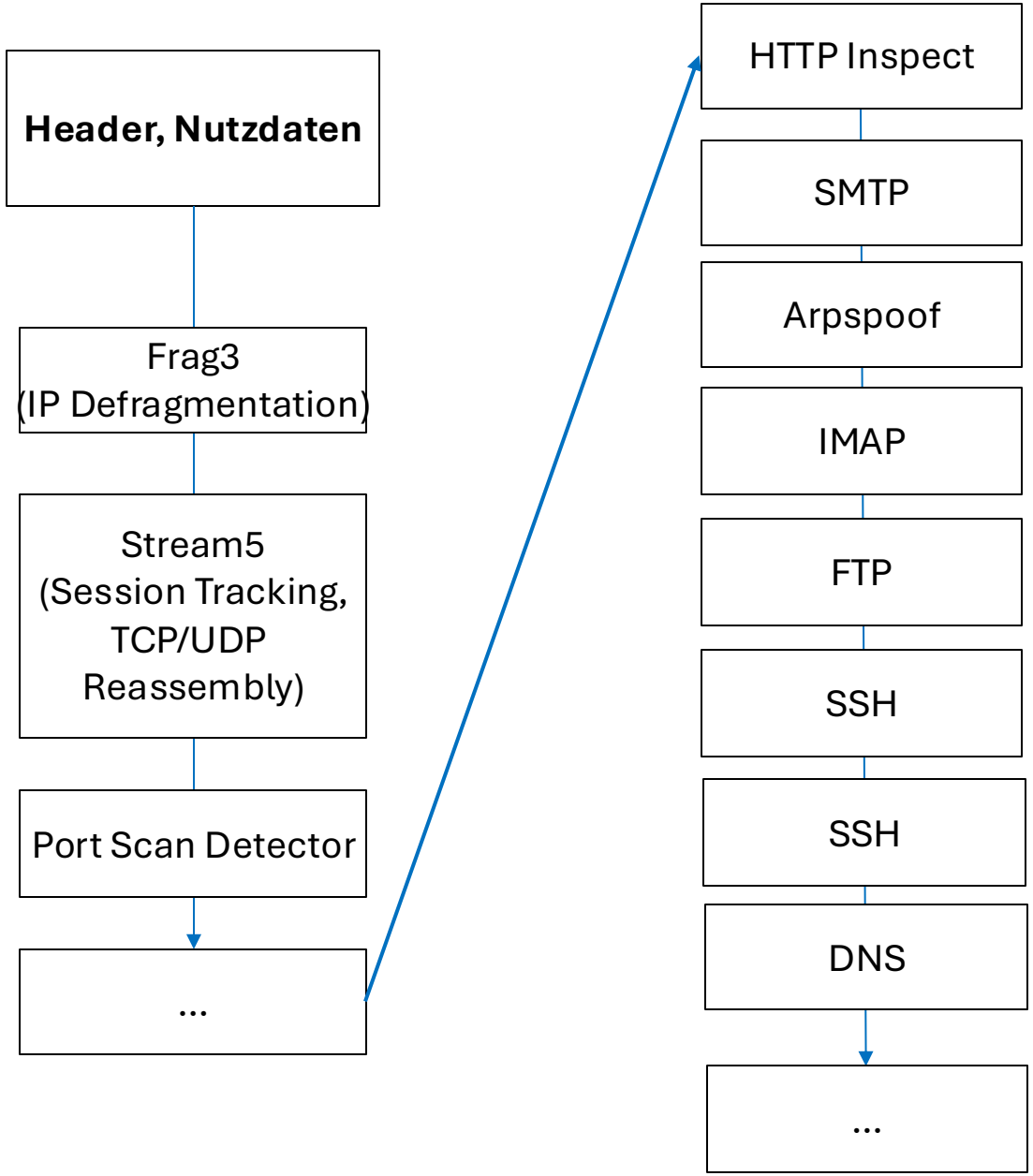
- Netzwerkkarte des Snort-Hosts muss im Promiscuous-Mode konfiguriert werden und liest dann alle empfangenen Netzwerkpakete unabhängig von der MAC-Empfängeradresse.
- Packet Decoder von Snort greift die Netzwerkdaten direkt von der Netzwerkkarte in ns-Geschwindigkeit mittels der
  - libpcap-Schnittstelle (LINUX) oder
  - winpcap-Schnittstelle (Windows)

  und reicht diese im pcap-Datenformat weiter.
- Beide Schnittstellen sind Open Source.
- Packet Decoder ermittelt im pcap-Datenstrom zuerst das Ethernet Frame und extrahiert die MAC-Adressen sowie das nachfolgende Protokoll (Feld: Ethertype) aus dem Ethernet-Header.



Snort Preprocessor

- Die Analyse der Protokoll-Header und der Applikations-Nachrichten erfolgt dann in der Präprozessorschicht.
- Um eine Vielzahl an Erweiterungen und Anpassungen an aktuelle Entwicklungen zu ermöglichen, werden die Präprozessoren in Form von SW-Plugins eingebunden.
- Aufgaben der Präprozessor-Schicht:
  - Zusammenfügen von IP-Fragmenten zu einem IP-Packet.
  - Zusammenfügen von TCP-Segmenten zu einem Applications-Bytestrom.
  - Analyse von Protocol-Misuse-Angriffen: TCP-Port-Scans (TCP-SYN-Request), ARP-Spoofing-Attacks, HTTP, ...
  - Auslesen von applikationsspezifischen Header-Feldern und Normalisierung des Nachrichteninhalts.



Beispiel: Stream5

Session Tracking verfolgt die Kommunikation zwischen zwei Endpunkten (z. B. Client und Server) über eine vollständige Sitzung hinweg.

- Session Tracking ist eine wichtige Funktion in Intrusion Detection Systems (IDS) wie Snort, da viele Angriffe sich nicht auf einzelne Pakete beschränken, sondern auf mehrere Pakete innerhalb einer Sitzung (Session) verteilt werden.
  - Ein Beispiel ist das Zusammenfügen (Reassembly) von fragmentierten TCP-Datenströmen, um vollständige Inhalte für die Analyse zu erhalten.
- Snort verwendet den STREAM5-Preprocessor für das Session Tracking.

- Stream5-Funktionsweise: Verfolgt TCP- und UDP-Verbindungen und stellt sicher, dass Snort vollständige Streams (nicht nur einzelne Pakete) analysieren kann.
  - TCP-Reassembly (Zusammenbau) von TCP-Segmenten mittels der Sequence-Nummer, um den vollständigen Datenstrom zu analysieren.
  - UDP-Tracking anhand der Quell- und Source-IP und dem Quell- und Source-Port.
  - Timeouts: Beendet Sessions, die inaktiv sind oder die maximale Dauer überschritten haben.
  - Session-Tracking für Anomalien: Erkennt ungewöhnliches Verhalten wie überlange Sessions, Pakete außerhalb der Reihenfolge.

Snort Rules

- Snort verwendet analog zu einer WAF ein Regelwerk zur Erkennung von Intrusion- oder Extrusion-Angriffen.
- Eine Snort-Regel besteht aus einem Rules Header und Rules Optionen

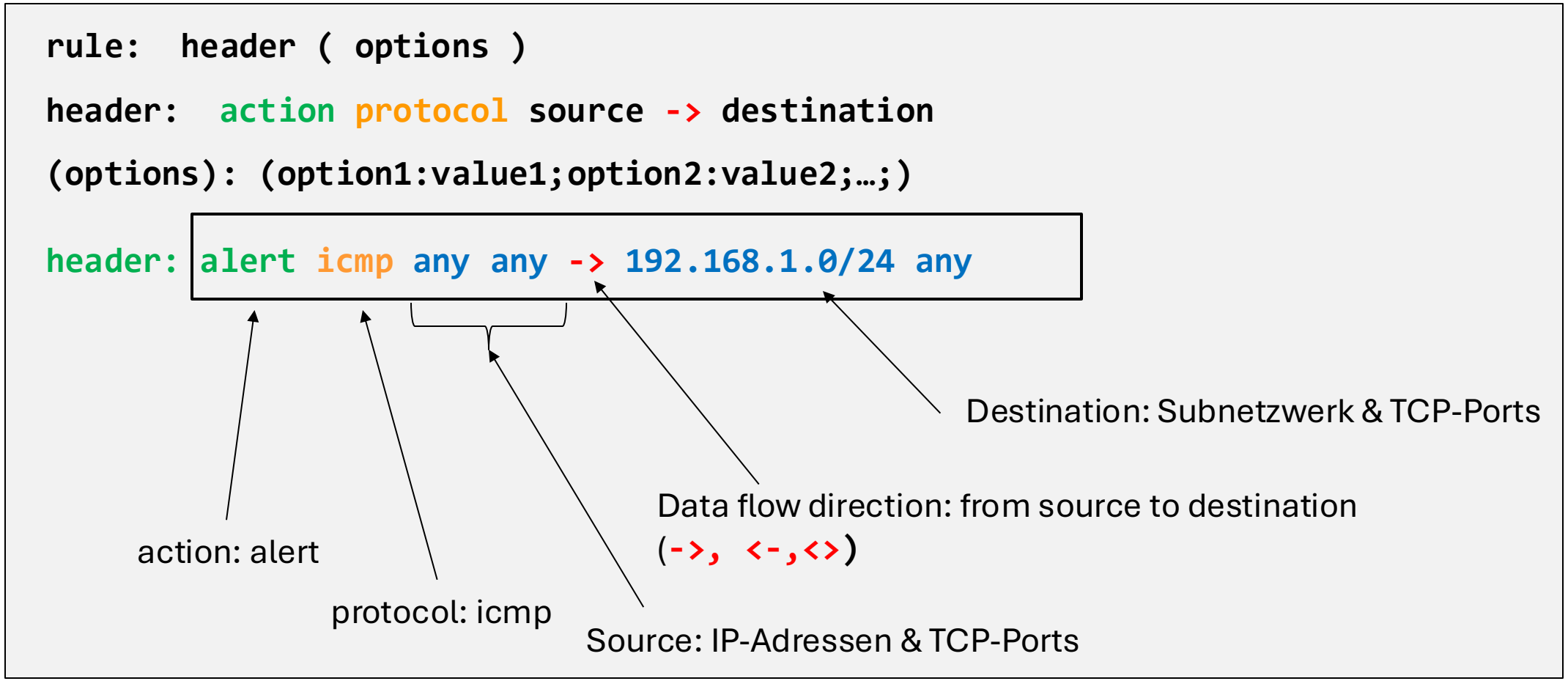
```
header [ ( options )]
```

```
header:    action protocol source -> destination
(options): (option1:value1;option2:value2;…;)
```

  - header: Definiert die Aktion (z. B. Alarm, Protokoll), das Protokoll, die Quell-/Ziel-IPs und die Ports.
  - options: Definiert die Bedingungen zum Auslösen der Regel, z. B. bestimmte Inhalte oder Bytemuster.
    Die Optionen werden durch eine runde Klammer umschlossen.

Snort – Regeln am Beispiel ICMP-Flooding Teil-1

- Generiere einen Alert wenn ICMP – Packete von `irgendwo (IP-Addr.== any ; TCP-Port==any)` in Richtung des Subnetzes `192.168.1.0/24` an `irgendwas (TCP-Port==any)` verschickt werden:

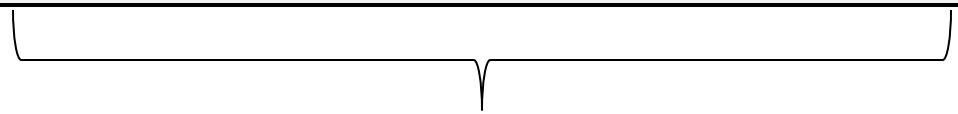


Snort – Regeln am Beispiel ICMP-Flooding Teil-2

```
(options): (option1:value1; option2:value2; …;)
```

```
(options): (msg:"ICMP flood"; detection_filter:track by_dst, count 500, seconds 3;)
```


msg: Nachricht für den Alert



detection_filter:

- track by_dst: verfolge pro Ziel-IP-Adresse.
- count 500: bei mehr als 500 ICMP Packete pro Beobachtungsintervall pro Ziel-IP-Addresse.
- seconds 3: Beobachtungsintervall 3s

Resultierende Regel:

```
alert icmp any any -> 192.168.1.0/24 any
(msg:“ICMP flood“; detection_filter:track by_dst, count 500, seconds 3;)
```

Snort Regeln

Die einzelnen Felder im Regel-Header können folgende Werte annehmen:

- Actions:

  `alert, pass, log, drop, …`

- Protocol:

  `tcp, udp, icmp, ip, arp`

- Directions

  `->, <-  (unidirectional)`

  `<>      (bidirectional)`

- Im Header können auch Variablen verwendet werden. Diese werden in der zentralen snort.conf - Datei definiert.

snort.conf:

```
var HOME_NET [192.168.1.0/24, 10.0.0.0/8]
var EXTERNAL_NET any
```

header: `alert tcp $EXTERNAL_NET any -> $HOME_NET 25`

Die folgenden Optionen können in Snort-Rules verwendet werden:

- Meta-Data:

  `msg: "Enthält auszugebende Nachricht"`

- Payload:

  `content: "Inhalt in Payload"`

  `pcre: "perl compatible regular expression"`

  `dsize:<300     (Payload kleiner 300 Byte)`

- Non-Payload:

  `flags:S        (TCP Syn Flag muss gesetzt sein)`

- Post-Detection:

  `logto: "filename" (Packete die eine Regel auslösen werden in ein File namens "filename" gespeichert.`

SNORT – IDS und IPS

Snort als IDS-System im out-of-band-Modus: Generiert einen Alert bei Regelübereinstimmung

```
alert icmp any any -> 192.168.1.0/24 any
(msg:"ICMP flood"; detection_filter:track by_dst, count 500, seconds 3;)
```

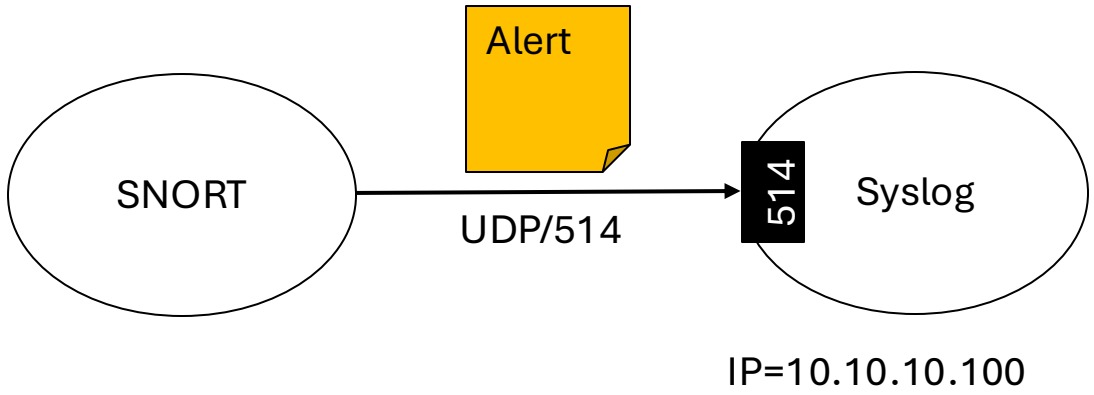
Tauscht man in obiger Regel die Aktion alert durch die Aktion drop aus, verwirft Snort automatisch die ankommenden ICMP-Packete und wirkt somit als IPS-System.

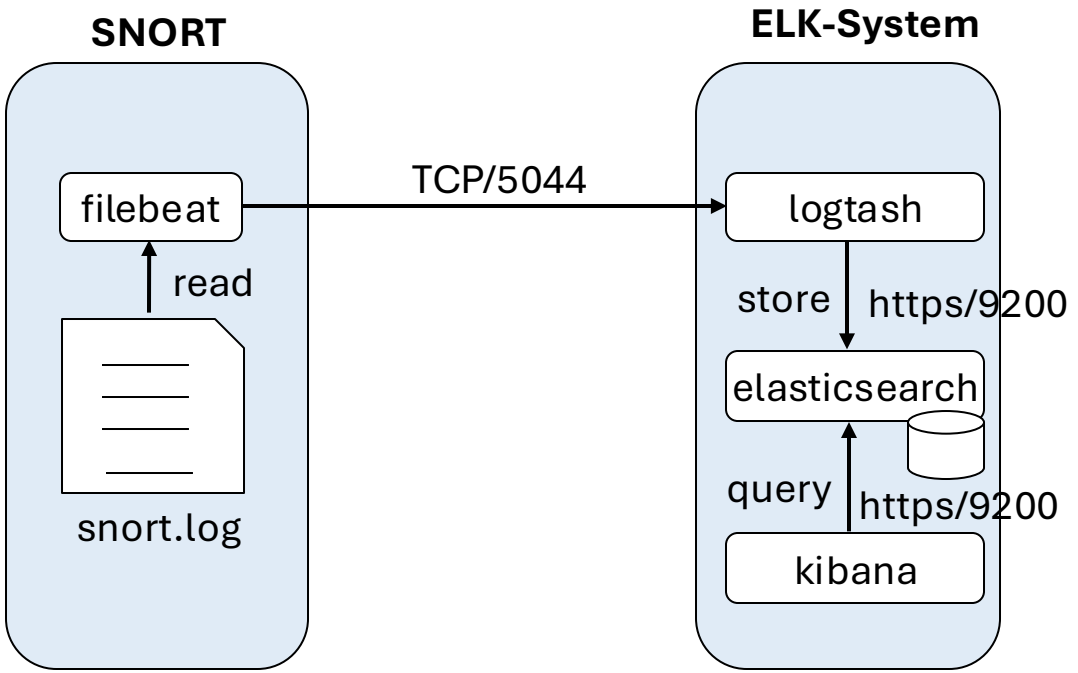
Snort als IPS-System im inline Modus: Generiert einen Alert und verwirft ICMP- Packet.

```
drop icmp any any -> 192.168.1.0/24 any
(msg:"ICMP flood"; detection_filter:track by_dst, count 500, seconds 3;)
```

SNORT: Output Module

- Die von SNORT generierten Alerts können mittels des Syslog-Protokolls auf einen zentralen Syslog-Server gespeichert werden.
  - Das Syslog-Protokoll verwendet UDP/514, um die Log-Nachrichten von einem Programm zu einem Syslog-Server zu schicken.
- Eine andere Möglichkeit besteht in der Weiterleitung der SNORT-Logs an ein zentrales ELK-System.
  - Installation eines Logdaten Shippers, z.B.: `filebeat,` der die in einer Logdatei gespeicherten Alerts zu einem zentralen Server schickt.
  - Serverseitig nimmt logtash die Daten entgegen, trans-formiert die Daten und sendet Sie zu elasticsearch.
  - elasticsearch indiziert und speichert die Daten.
  - kibana liest Daten und visualisiert Sie in Form von Dashboards.





SNORT und ML

- Eine Kombination von **Snort** mit einem Machine Learning (ML) Verfahren, ermöglicht eine Analyse der von Snort gesammelten Daten mit dem Ziel unbekannte Angriffsmuster zu erkennen.
- Die so erkannten neuen Angriffe können dann durch neue Regeln erkannt und blockiert werden.
- **1 Paketweiterleitung (Snort → ML-System)**:
  - Snort im IPS-Modus inspiziert ankommende Pakete.
  - Weiterleitung von Paketen oder Logs an ein ML-System, z. B. über Syslog oder JSON-API oder via Direct Streaming (z. B. Apache Kafka).
- **2 Echtzeit-Analyse durch ML**:
  - Implementiere ein ML-System, das in Echtzeit von Snort erhaltenen Netzwerkverkehr bewertet.
  - Anomalieerkennung: Vergleich von aktuellem Netzwerkverkehr mit normalem Verhalten (baseline).
  - Klassifizierung: Einteilen von Paketen in “bösartig” oder “harmlos”.
- **3 Integration der Ergebnisse in Snort**:
  - Blockiere verdächtige Pakete mit neuen Snort-Regeln (z. B. über dynamische Regelupdates).
- **Tools zur Unterstützung:**
  - Apache Kafka: Für die Datenweiterleitung von Snort zu einem ML-System.
  - ELK-Stack + ML-Plugin: Visualisierung und Machine-Learning-basierte Mustererkennung.
  - Scikit-learn oder TensorFlow: Zur Implementierung von Echtzeit-ML-Modellen in Python.

Aufgabe 2.1: FW- und IPS-Regeln

a) Konfigurieren Sie das äußere Interface (eth1/2) einer **ASA**-Edge-Paket-Firewall ("outside", 0, 10.0.0.2) und definieren Sie eine ACL die smtp-Nachrichtenverkehr aus dem Internet auf den Host mit der IP-Adresse 192.168.56.25 erlaubt.

b) Die WAF **Modsecurity** kann dazu verwendet werden, Teile ihrer Applikation, die eine Schwachstelle besitzen temporär vor Zugriffen zu schützen. Erstellen Sie eine Regel die den Zugriff auf die API /cgi/get_saldo.sh blockiert. Die Severity der Regel soll 2 betragen.

c) Erstellen Sie eine **SNORT-Regel** die eine SQL-Injection (OR 1=1 --) in einem Netzwerkpaket erkennt und dieses Paket blockiert.

d) Erstellen Sie eine **SNORT-Regel** die einen Gratituous-ARP Angriff erkennt und diesen meldet. Erklären Sie die Regel.

Aufgabe 2.2: Firewalls und IPS/IDS-Systeme

1. **Grundlagen**
   a. Was ist der Unterschied zwischen **Intrusion Detection Systemen (IDS)** und **Intrusion Prevention Systemen (IPS)?**
   b. Was ist der Unterschied zwischen einer **Packet-**, einer **Stateful-FW** und einer **Next-Generation-Firewall**?
   c. Welche Hauptfunktionen erfüllt ein IPS und welche Hauptfunktionen erfüllt eine FW in einem Netzwerk?
2. **Funktionsweise und Methoden**
   a. Beschreiben Sie die grundlegende **Arbeitsweise eines IPS**.
   b. Welche **Detektionsmechanismen** werden in einem IPS eingesetzt? Erklären und vergleichen Sie **Signaturbasierte Erkennung, Anomaliebasierte Erkennung** und **Verhaltensbasierte Erkennung**.
3. **Vergleich und Herausforderungen**
   a. Welche Vorteile bietet ein IPS gegenüber einer klassischen Firewall?
   b. Welche **Herausforderungen und Limitierungen** gibt es bei der Implementierung eines IPS?
   c. Was sind **False Positives** und **False Negatives**?
   d. Warum kann ein **IPS allein kein vollständiger Schutz** für ein Netzwerk sein? Welche weiteren Sicherheitsmaßnahmen sind notwendig?
4. **Angriffsszenarien & Schutzmaßnahmen**
   a. Nennen und beschreiben Sie drei unterschiedliche Angriffe, gegen die ein IPS-System Schutz bieten kann.
   b. Wie kann ein Angreifer versuchen, ein IPS zu umgehen?