IP Security

Zwei Mechanismen:

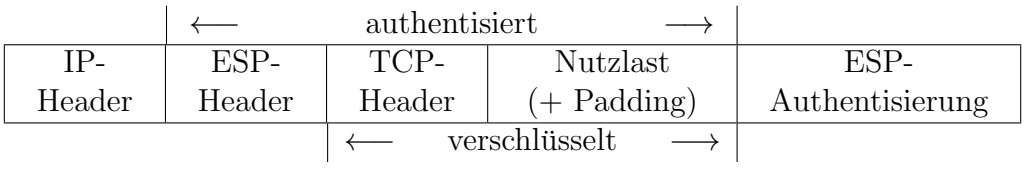
- Authentication Header: Nur Datenauth. (Exportbeschränkungen)
  Empfehlung: Nicht mehr umsetzen
- Encapsulating Security Payloads (ESP): Verschl., Datenauth.

Internet Key Exchange Protokoll: Schlüsselaustausch

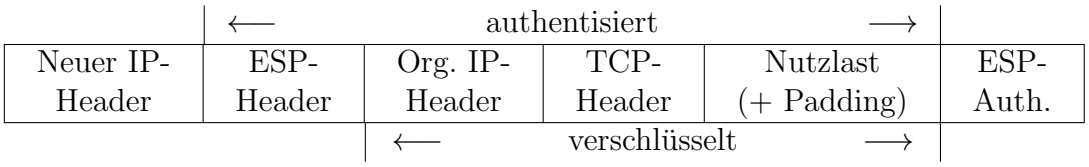
- Mit und ohne Instanzauthentisierung
  (über Zertifikate oder Pre Master Keys (vorab ausgetauscht))
- Generierung von Security Associations (SA)
  Für Kommunikation $A \to B$ und $B \to A$
- Inhalt SA: Schlüssel (für Versch. + Auth.), IV, Algoritm Identifier
- SAs sind identifiziert über Security Parameter Index (SPI, 32 Bit)

ESP: Verschlüsselung und Datenauthentisierung (zwei Modi)

- Transportmodus: Direkte Verbindung von Host zu Host



- Tunnelmodus: Verbindung zwischen Security Gateways (z.B. für VPN)



- Inhalte im ESP-Header: SPI, Sequenznummer

19.2 Angreifertyp 2: Netzwerksicherheit

Erinnerung: Angriffe auf lokales Netzwerk
Angreifer ist ein bösartiger Knoten außerhalb des lok. Netzes

Beispiel. Mit Remote Login (rlogin) host-based Autentisierung möglich.

- Nutzer ist bereits in einem vertrauenswürdigen Rechner eingeloggt
- Einloggen ohne Passwort (unter selbem Nutzer) möglich

Festlegung in der Datei `/etc/hosts.equiv` (root) oder `rhosts` (Nutzer)

- Verbindung zwischen beiden Rechnern (C und S) über TCP
- Verbindungsaufbau (TCP-Handshake):
  - $C \to S$: SYN, ISS_C (32 Bit Seq.-Nummer)
  - $S \to C$: SYN|ACK, ISS_S, ACK(ISS_C) :=$\text{ISS}_C + 1$
  - $C \to S$: ACK, $\text{ISS}_C + 1$, ACK(ISS_S)

  SYN (Synchronize), ACK (Acknowledgment): Flags im TCP-Header
- Verbindung: ISS_C, ISS_S sind Paketnummern, Empfänger quittiert (ACK)
- Verbindungsabbau: C oder S sended FIN-Flag

Ziel des Angreifers A außerhalb des lok. Netzes (TCP Session Hijacking):

- Senden eines Befehls über rlogin auf S mit Privilegien eines Nutzer U
- Nebenbedingung: U ist bereits auf C eingeloggt

Vorgehen:

- Angreifer A beginnt Handshack (mit Adresse von C): $A(C) \to S : \text{ISS}_A$
- Problem 1: Antwort wird von S zu C geschickt (A kennt ISS_S nicht)

- Lösung: ISS ist i.A. nicht zufällig (wird einfach hochgezählt)
  Bsp. Berkley Unix: Jede Sekunde um 128, für jeden Versuch um 64
- Angreifer baut zunächst reguläre Verbindung zu S auf
  - $A \to S : \mathrm{ISS}_A$; $S \to A : \mathrm{ISS}_S$, $\mathrm{ACK}(\mathrm{ISS}_A)$
- Danach sofort $A(C) \to S : \mathrm{ISS}_{A(C)}$; $S \to A : \mathrm{ISS}'_S$, $\mathrm{ACK}(\mathrm{ISS}_{A(C)})$
  A kann ISS'_S mit hoher Wkeit erraten (über Kenntnis von ISS_S)
- Problem 2: C erhält Antwort auf ein Paket (und sendet FIN-Flag)
  S akz. damit keine auf ISS'_S basierenden Pakete mehr)
- Lösung: TCP Syn Flooding (typische DoS-Attacke)
  - A sendet SYN-Anfragen an C ohne auf Cs ACK zu antworten
  - A hält alle Verbindungsanfragen offen (bis Maximum erreicht ist)
    ACKs von Ss werden nicht mehr verarbeitet (kein senden der FIN)

Verhinderung des Angriffs:

- Keine Verwendung von host-based Authentisierung
- Krypt. Vernetzung auch im lok. Netz (Kerberos)
- Firewalls: IP-Pakete von außen mit int. Adressen blockieren
- Intrusion Detection: Erkennen von Anomalien

Firewall-Technolgien

Grundidee: Gesamter Datenverkehr zw. innen (lok. Netz) und außen (Internet) läuft über eine Firewall

- Zugriffe können kontrolliert werden (Sicherheitsstrategie)
- Zugriffe können protokoliert werden

Wir unterscheiden (werden häufig kombiniert):

- Paketfilter, Zustandsgesteuerte Filter, Proxiy-Filter, Applikationsfilter

Paketfilter: Angesiedelt auf IP- und Transportlayer
Entscheidung an Hand der IP- und TCP-Header: Absender (Adresse, Port), Empfänger (Adresse, Port), Protokoll (http, smtp, ...)

Sicherheitsstrategie: Festlegung über Tabelle:

Beispiel. Auszug aus einer Sicherheitstabelle

Aktionen	IP-Adr. Abs.	Port Abs.	IP-Adr. Empf.	Port Empf.	Protokoll Appl.-Layer	Bedeutung
blockieren	intern	$\star$	intern	$\star$	$\star$	Bsp. oben
blockieren	PC Pool	$\star$	$\star$	$\star$	$\star$	Kein Zugriff nach außen
erlauben	intern auth.	80	$\star$	80	http	

($\star$ = alle)

- Vorteil: Einfach umsetzbar (auch in Routern mit beschr. Ressourcen)
- Nachteil: statische Tabelle, Nutzlast wird nicht analysiert

Zustandsgesteuerte Filter: Angesiedelt auf IP- und Transportlayer
Beispiel. Client C möchte via http auf Server S zugreifen

- Zulassen von Paketen $S \xrightarrow{http} C$ nur, wenn vorab $C \xrightarrow{http} S$

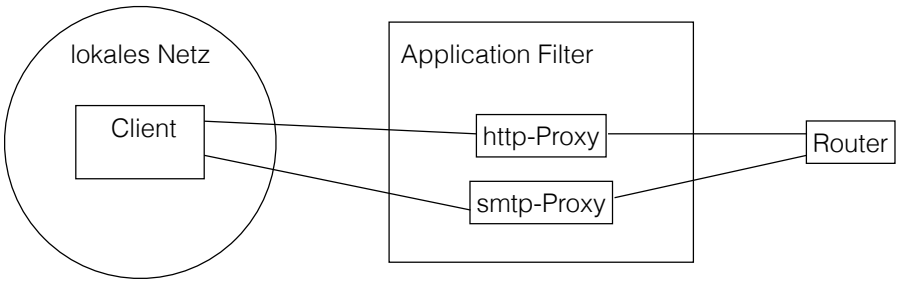
Proxy-Filter (Stellvertreter): Angesiedelt auf Transport Layer
Beispiel. Client C will Server S kontaktieren

- Proxy P tritt gegenüber S als Client C auf
- und gegenüber C als Server S

- Vorteil:
  - Client muss keine Sicherheitstrategie umsetzen (erlaubt nur interne Kommunikation)

  - Umfangreiche Regeln umsetzbar (auch Analyse Nutzlast)
- Nachteile: Komplex und damit selbst Ziel von Angriffen

Application-Filter: Angesiedelt auf Schicht 7 (Application Layer)
Analyse der Nutzlast nach bekannten Angriffen (Viren, Würmer, ...)

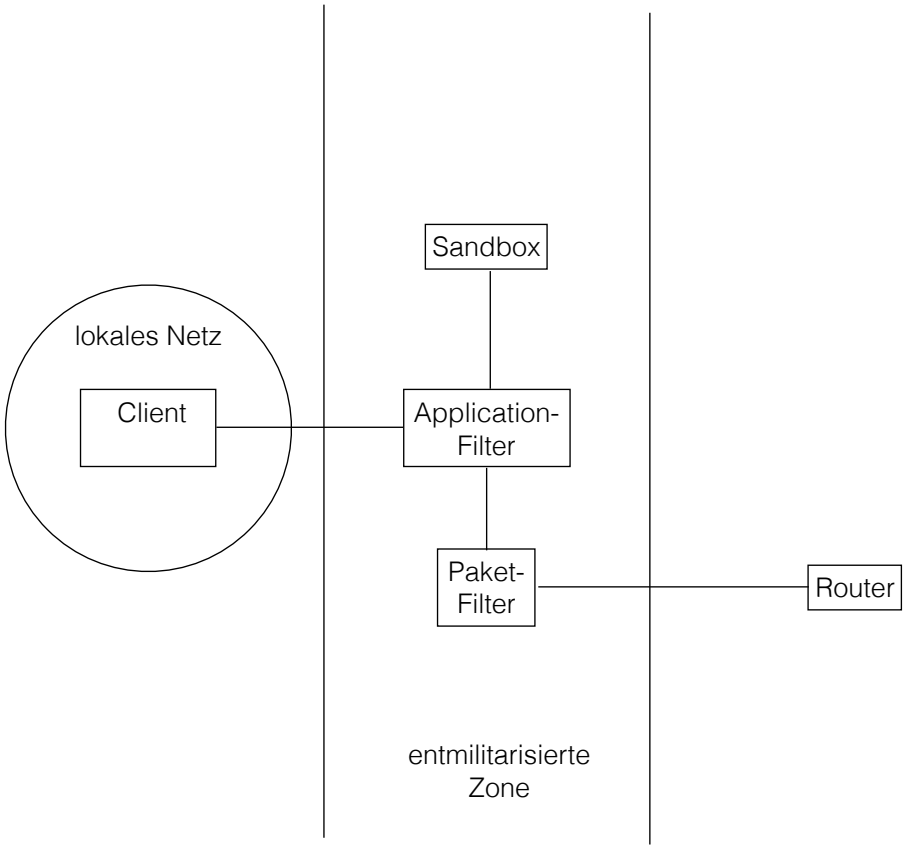


- Vorteil: Deutlich bessere Analyse möglich
- Nachteil: Komplex, daher selbst eine potentielle Schwachstelle

Lösung:

- Analyse der Nutzlast in gesicherter Umgebung (Sandbox)
- Absicherung des Appl.-Filters durch andere Firewalls

Entmilitarisierte Zone (Demilitarized Zone, DMZ)


Sandbox
lokales Netz
Client
Application-
Filter
Paket-
Filter
Router
entmilitarisierte
Zone