Domain Name System (DNS)

Hauptfunktion Namensauflösung: google.de → 173.194.112.111

Beispiel (Auflösung von google.de).

- Client → Resolver: Auflösung google.de
- Resolver → Rootserver: Liefert Toplevel Domain Server (TLD)
- Resolver → TLD: Liefert autoritativen DNS-Server
- Resolver → DNS: Liefert 173.194.112.111
  Resolver cached Antwort (mit Flag Time To Life, TTL)
- Resolver → Client: 173.194.112.111

Abfrage über UDP/IP, Antwort u.a. IP-Adresse (A Record (IPv4), AAAA Record (IPv6))
Absicherung: 16-Bit Query-ID (wird jeweils zurückgeliefert)

DNS Cache Poisoning Attack

Ziel: Zurückliefern einer falschen IP-Adresse für abgefragten Host

- z.B. zur Durchführung von Phishing-Attacken
- www.paypal.de führt zu www.attacker.de

Attacke:

- Angreifer A bringt Client C dazu, Adresse von paypal abzufragen
  z.B. über Internetseite mit Inhalt
- A überflutet DNS-Server von C mit falschen IP-Adressen (QID geraten)
- Kommt die Antowrt von A zuerst an, wird diese weitergeleitet
- Erfolgsausicht:
  - Nur Raten von QID und Portnummer (kein Handshake bei UDP)
  - Wkeit für QID: $1/2^{16}$ (Besser mit mehr Antworten durch A)

  - Wkeit für Port: Häufig statisch, also 1
- Gegenmaßnahme: Zufällige Portnummern

Problem: Antwortet DNS zuerst, neuer Angriff erst nach Ablauf TTL

Dan Kaminsky-Attacke:

- DNS-Server können auch weitere IP-Adressen liefern (Glue Records)
- Bailiwick Checking: Akz. der Antwort nur, wenn im selben Bereich
  Adr. für 123.example.com (A Rec.), Adr. für example.com (Glue Rec.)
- DNS-Server cashed dann auch glue record (unabhängig von TTL)
- Angriff:
  - C fragt Adressen 111.paypal.de, 112.paypal.de, ... ab
  - A überflutet C mit gefälschten Anworten + glue record für paypal.de
- Antwort muss ankommen, bevor DNS antwortet (mit NXDOMAIN)
- Wiederholung, wenn DNS vorher antwortet (Dauer ca. 10 Sekunden)

Gegenmaßnahme: (Übung)

- Split-Split DNS-Server (Einführung nach Bekanntwerden der Attacke)
- DNSSec (krypt. Absicherung, noch in Einführung)

Firewall-Technolgien

Idee: Datenverkehr zwischen lok. Netz und Internet läuft über eine Firewall

- Zugriffe können kontrolliert und protokolliert werden

Wir unterscheiden (werden häufig kombiniert):

- Paketfilter, Zustandsgesteuerte Filter, Proxy-Filter, Applikationsfilter

Paketfilter: Angesiedelt auf IP- und Transportlayer
Entscheidung an Hand der IP- und TCP-Header: Adressen, Ports

Sicherheitsstrategie: Festlegung über Tabelle:

Beispiel. Auszug aus einer Sicherheitstabelle

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

($\star$ = 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

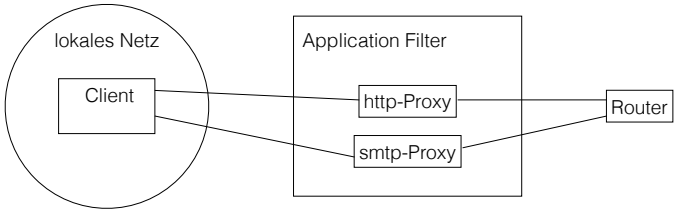
- Zulassen von Paketen $S \xrightarrow{http} C$ nur, wenn vorab $C \xrightarrow{http} S$

Weiterl. von TCP-Pakete nur, wenn Client TCP-Handshake initiiert hat

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

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

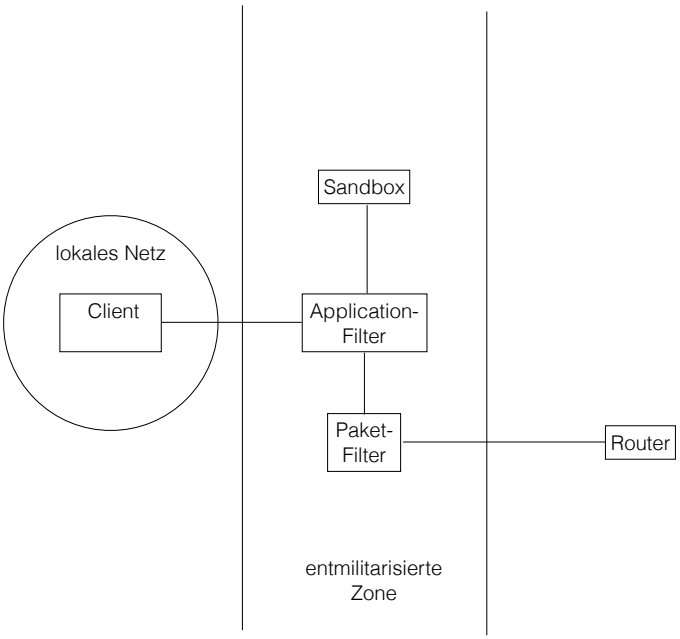


- Vorteil:
  - Client muss keine Sicherheitstrategie umsetzen (erlaubt nur interne Kommunikation)
  - Umfangreiche Regeln umsetzbar (auch Analyse Nutzlast)
- Nachteile: Komplex und damit selbst Ziel von Angriffen

Lösung:

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

Entmilitarisierte Zone (Demilitarized Zone, DMZ)



Intrusion Detection System: Erkennung aktuell laufender Angriffe

- Misuse Detection (z.B. häufig fehlgeschlagene Login-Versuche)
- Angriffe hinerlassen häufig Spuren (Angriffssignaturen)
  - Netzwerkbasierte IDS, z.B.
    - Analyse Nutzlast von TCP-Paketen nach bekannten Exploids
    - Analyse von TCP-Headern (Erkennen von floodings)
  - Hostbasierte IDS, z.B.
    - Suche von Angriffsignaturen in Logfiles
    - Checksummenprüfung der wichtigen Systemdateien
- Anomaly Detection (z.B. Login zu seltsamer Uhrzeit)
  - Benutzung statistischer Techniken (Abweichung vom Normalfall)
  - Justierung zwischen false positiv und false negativ nötig

Dazu wichtig: Kenntnisse über Verwundbarkeiten (z.B. aktuelle Angriffe)

- Computer Emergency Response Teams (CERT), z.B. www.bsi.de
- Honeypots: Vortäuschung echter Systeme zur Analyse von Angriffen