1 Domain Name System

Das **Domain Name System (DNS)** ist ein fundamentaler Dienst des Internets, oft bezeichnet als das „Telefonbuch des Internets“. Es wurde ca. 1985 entworfen (Ursprung im ARPANET) und ursprünglich **ohne** Sicherheitsfeatures konzipiert.

Kernfunktion

Das DNS ist eine **globale, verteilte Datenbank**, die hierarchisch verwaltet wird. Die Hauptaufgabe ist die Übersetzung (Auflösung) von menschenlesbaren Hostnamen (z. B. `www.example.org`) in maschinenlesbare IP-Adressen (z. B. `93.184.216.34`).

1.1 Hierarchie und Namensraum

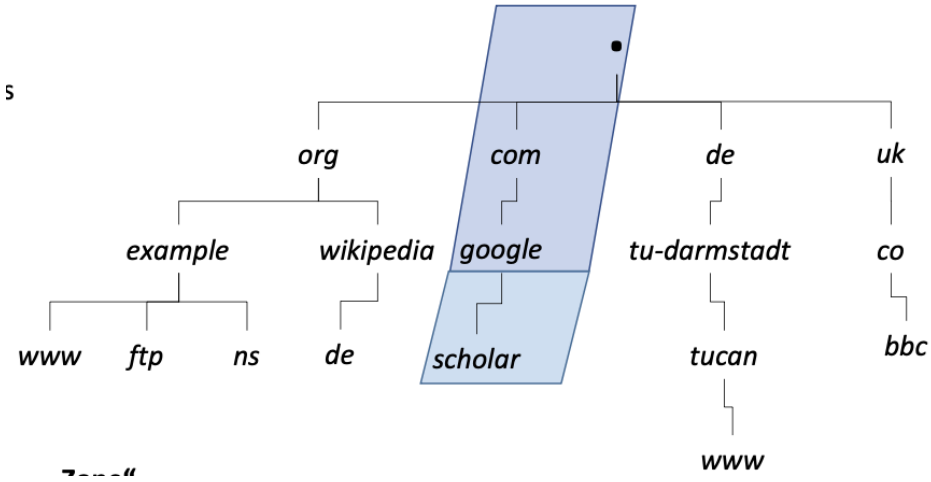
Das DNS ist als Baumstruktur organisiert:

- **Root (.):** Die Wurzel des Baums (oft als Punkt am Ende dargestellt).
- **Top-Level-Domains (TLDs):** Unterhalb der Root (z. B. `org`, `com`, `de`).
- **Second-Level-Domains:** Z. B. `example` in `example.org`.
- **Subdomains:** Weitere Unterteilungen, z. B. `www` oder `ftp`.

1.2 Domain vs. Zone

Es ist wichtig, zwischen einer logischen Domain und einer administrativen Zone zu unterscheiden.

- **Domain**: Ein logisch abgegrenzter Teilbereich des Internets mit eindeutigem Namen.
- **Zone**: Ein von einer **einzigen Autorität** verwalteter Bereich. Eine Zone kann eine Domain umfassen, schließt aber Subdomains aus, die an andere Autoritäten delegiert wurden (z. B. wird eine Subdomain administrativ ausgegliedert, bildet sie eine eigene Zone).



1.3 DNS Resource Records (RR)

Informationen im DNS werden in sogenannten **Resource Records** gespeichert. Ein Record besteht aus folgenden Feldern:

1. **Name (Owner):** Identifikator (FQDN - Fully Qualified Domain Name).

2. **Type:** Art des Datensatzes (siehe Tabelle).
3. **Class:** Meist `IN` (Internet).
4. **TTL (Time to Live):** Gültigkeitsdauer in Sekunden (für Caching).
5. **RDLength:** Länge der Daten in Bytes.
6. **RData:** Der eigentliche Wert (z. B. die IP-Adresse).

1.3.1 Wichtige Record-Typen

Typ	Beispiel-Daten	Zweck
A	93.184.216.34	IPv4-Adresse zum Hostnamen.
AAAA	2606:2808::1	IPv6-Adresse zum Hostnamen.
MX	mail.example.org	Mail Exchange: Mailserver für die Domain.
NS	ns.example.org	Name Server: Autoritativer Server für eine Zone.
CNAME	server1.blau.de	Canonical Name: Alias auf einen anderen Namen.
TXT	v=spf1 -all	Beliebiger Text (oft für Sicherheitsmechanismen wie SPF).

RRset

Ein **Resource Record Set (RRset)** ist die Menge aller Records mit **gleichem Namen, Typ und Klasse**. DNS überträgt immer ganze RRsets, nie einzelne Records aus einem Set (z. B. beim Load-Balancing mit mehreren A-Records für eine Domain).

1.4 Nachrichtenübermittlung und Auflösung

1.4.1 Kommunikation

DNS verwendet ein Client/Server-Modell.

- **Transport:** Standardmäßig **UDP Port 53**. TCP Port 53 wird bei großen Antworten (Zone Transfers, große DNSSEC-Pakete) verwendet.
- **Format:** Anfragen und Antworten haben dasselbe Format (Header, Question, Answer, Authority, Additional).

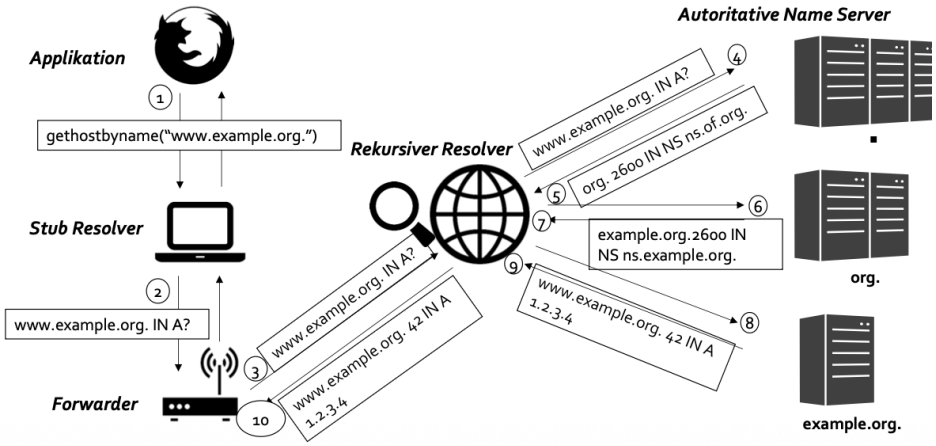
1.4.2 Server-Typen und Rollen

- **Stub Resolver**: Einfacher Client auf dem Endgerät (PC/Laptop), stellt nur Anfragen.
- **Forwarder**: Leitet Anfragen weiter (z. B. Router im Heimnetz).
- **Rekursiver Resolver**: "Der Suchenende". Übernimmt die komplette Auflösung für den Client, fragt verschiedene Server ab und cacht Ergebnisse (z. B. Server beim ISP oder Google 8.8.8.8).
- **Autoritativer Name Server**: "Der Wissende". Hat die Hoheit über eine Zone und liefert die endgültigen Antworten.

1.4.3 Auflösungsmethoden

Rekursive vs. Iterative Anfragen

- **Rekursiv (RD-Flag=1):** „Besorge mir die Antwort." Der angefragte Server übernimmt die Arbeit und liefert das Endergebnis. Typisch zwischen *Stub Resolver* und *Rekursivem Resolver*.
- **Iterativ (RD-Flag=0):** „Gib mir die Antwort oder sag mir, wen ich fragen soll." Der Server liefert entweder die Daten oder einen Verweis (Referral) auf den nächsten zuständigen Server. Typisch zwischen *Rekursivem Resolver* und *Autoritativen Servern*.



1.5 DNS Cache Poisoning

Da DNS ursprünglich keine Authentifizierung besaß, vertrauen Resolver den Antworten, die sie erhalten (UDP ist verbindungslos und leicht zu fälschen).

Cache Poisoning

Einspeisung gefälschter DNS-Einträge in den Cache eines Resolvers. Ziel ist meist die **Impersonation** (Umlenkung von Nutzern auf Angreifer-Server).

1.5.1 Angriffsmethoden

1. **Klassisches Poisoning (Pre-Bailiwick):** Angreifer sendet Antwort mit zusätzlichen, gefälschten Records für fremde Domains (z. B. „Hier ist die IP für `example.org`, und übrigens ist die IP für `google.com` 6.6.6.6“).
2. **Off-Path Angriff:** Der Angreifer kann den Verkehr nicht mitlesen (Blind spoofing). Er muss die Anfrage des Resolvers an den autoritativen Server erraten und schneller antworten als der echte Server.
   - *Herausforderung:* Erraten der 16-Bit **Transaction-ID** und des **UDP-Quellports**.
3. **Kaminsky Angriff (2008):** Ein ausgeklügelter Off-Path Angriff. Um das Problem zu umgehen, dass ein Cache-Eintrag (selbst ein fehlgeschlagener) eine TTL hat und weitere Angriffsversuche blockiert:
   - Angreifer fragt nicht-existente Subdomains an (z. B. `1.bank.com`, `2.bank.com`).
   - Resolver muss jedes Mal neu beim autoritativen Server fragen.
   - Angreifer flutet gefälschte Antworten, die einen neuen, gefälschten Nameserver für die Ziel-Zone (`bank.com`) einschmuggeln.
   - *Gegenmaßnahme:* Randomisierung des UDP-Quellports (zusätzlich zur Transaction-ID), was den Suchraum auf ca. 32-Bit erhöht.
4. **Man-in-the-Middle (MITM):** Angreifer kann Verkehr mitlesen (z. B. im WLAN oder via BGP-Hijacking). Transaction-ID und Ports sind sichtbar → Triviales Poisoning möglich.

1.5.2 Bailiwick-Regel (Gegenmaßnahme)

Ein Resolver akzeptiert nur Informationen, die in den Zuständigkeitsbereich (Zone) des antwortenden Servers fallen.

- Ein Server für `example.org` darf keine Records für `google.com` liefern.
- Er darf aber Records für `www.example.org` liefern.

1.6 Gegenmaßnahme: DNSSEC

Um MITM und fortgeschrittenes Cache Poisoning zu verhindern, muss die Authentizität der Daten sichergestellt werden. **DNSSEC** (DNS Security Extensions) bietet Integrität und Authentizität durch kryptographische Signaturen, aber **keine** Vertraulichkeit (Daten sind lesbar).

1.6.1 Funktionsweise

DNSSEC bildet eine **Chain of Trust** von der Root-Zone bis zur Ziel-Domain.

- Records werden nicht verschlüsselt, sondern **signiert**.
- Eltern-Zonen signieren den Hash der Schlüssel ihrer Kinder (Delegation).

1.6.2 Neue Record-Typen

- **RRSIG**: Enthält die digitale Signatur eines RRsets.
- **DNSKEY**: Enthält den öffentlichen Schlüssel (Public Key) zum Überprüfen der Signatur.
- **DS** (Delegation Signer): Fingerprint (Hash) des Schlüssels der Unterzone (liegt in der Elternzone, stellt die Vertrauenskette her).
- **NSEC / NSEC3**: Dient dem *Authenticated Denial of Existence* (Beweis, dass ein Name **nicht** existiert).

1.6.3 Problem: Zone Enumeration

Da DNSSEC beweisen muss, dass ein Name *nicht* existiert, geben NSEC-Records Informationen über den „nächsten“ existierenden Namen preis.

- **NSEC Walking:** Angreifer fragt nacheinander Namen ab und erhält durch die NSEC-Antworten („Zwischen A und F gibt es nichts“) die Liste aller existierenden Domains.
- **NSEC3:** Hasht die Namen. Angreifer können die Hashes jedoch offline via Brute-Force (GPU) knacken, da der Namensraum (z. B. www, mail) klein ist.
- **Lösung (Live Signing / White Lies):** Server berechnet Signaturen on-the-fly. Bei Anfrage nach `ghost.example.com` behauptet der Server: „Der Vorgänger ist `ghost` und der Nachfolger ist `ghost\000`“. Der Bereich ist so klein, dass er nur den angefragten Namen abdeckt. Verhindert Enumeration effektiv.

1.7 Transport-Sicherheit (Privacy)

DNSSEC schützt die Daten, verschlüsselt aber nicht den Transport. Wer wissen will, welche Webseiten ein Nutzer besucht, kann den DNS-Verkehr mitlesen.

DoT vs. DoH

Beide Protokolle verschlüsseln die Kommunikation zwischen Stub-Resolver und Rekursivem Resolver (Last-Mile-Security).

- **DoT (DNS over TLS):** Dedizierter Port (TCP 853).
- **DoH (DNS over HTTPS):** Versteckt DNS im HTTPS-Traffic (TCP 443). Schwerer zu blockieren/-filtern.

Wichtig: Sie schützen vor Lauschern auf der Leitung, garantieren aber **nicht** die Echtheit der Daten vom autoritativen Server (dafür wird DNSSEC benötigt).

1.8 Weitere Angriffe auf DNS-Infrastruktur

1.8.1 DNS Amplification DDoS

Ein **Reflection**-Angriff unter Ausnutzung des UDP-Protokolls.

1. Angreifer sendet Anfrage an offene DNS-Server.

2. **IP-Spoofing:** Absender-Adresse ist die des Opfers.
3. **Amplification:** Die Anfrage ist klein (z. B. 60 Byte), die Antwort ist riesig (z. B. 3000 Byte, Faktor 50x).
4. Der DNS-Server flutet das Opfer mit den großen Antworten.

Gegenmaßnahmen: Response Rate Limiting (RRL) auf Servern, Verhinderung von IP-Spoofing im Netzwerk (BCP 38).

1.8.2 DNS Tunneling

Umgehung von Firewalls oder Exfiltration von Daten.

- Daten werden in Subdomains kodiert (z. B. `geheimespasswort.angreifer.com`).
- Der autoritative Server des Angreifers empfängt die Anfrage und dekodiert die Daten.
- Antworten können Steuerbefehle (C2) enthalten (via TXT oder CNAME Records).

1.9 DNS-basierte Sicherheitsmechanismen für E-Mail

Das DNS wird genutzt, um die Sicherheit anderer Dienste (v.a. E-Mail) zu erhöhen.

1.9.1 SPF (Sender Policy Framework)

Schutz gegen E-Mail-Spoofing (Versand unter falschem Namen).

- Ein **TXT-Record** in der Domain definiert, welche IP-Adressen Mails für diese Domain versenden dürfen.
- **Syntax:** `v=spf1 [Mechanismen] [Qualifier]all`

Qualifier	Bedeutung
`+`	Pass (Standard, wenn weggelassen).
`-`	Fail (Mail ablehnen).
`~`	Soft-Fail (Mail annehmen, aber markieren/Spam-Ordner).
`?`	Neutral.

Beispiel: `v=spf1 mx ip4:1.2.3.0/24 -all`
Bedeutet: Die MX-Server und das Subnetz 1.2.3.0/24 dürfen senden. Alles andere (`-all`) wird abgelehnt.

1.9.2 Weitere Mechanismen

- **DKIM**: Signieren von E-Mails; Public Key liegt im DNS.
- **DANE**: Bindung von TLS-Zertifikaten an DNS-Namen (via TLSA-Records), benötigt zwingend DNSSEC.