Kapitel 9: Domain Name System (DNS)

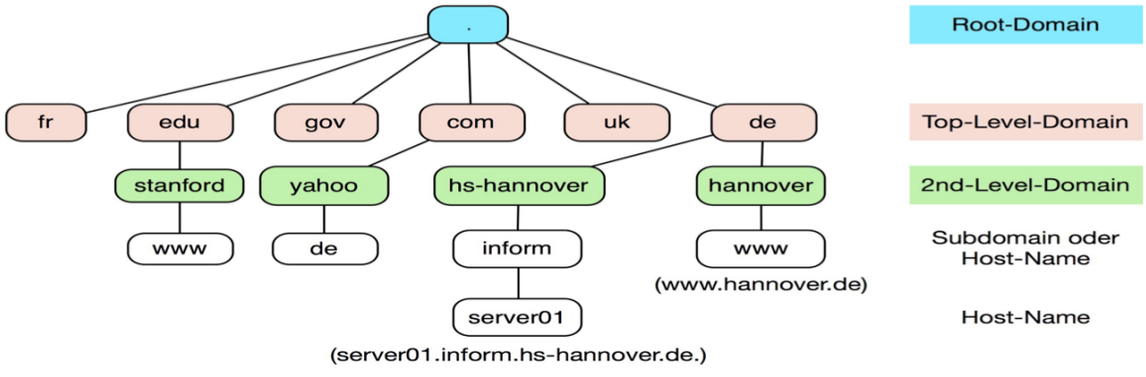
Motivation und Grundlagen von DNS

- IPv4-Adressen lassen sich nicht einfach merken. *Beispiele: 173.194.65.94, 141.71.30.206, etc.*
- IPv6-Adressen kann lassen sich kaum noch fehlerfrei eintippen.
- Namen lassen sich viel einfacher im Gedächtnis behalten.
- Durch zusätzliche Strukturierung des Namensraums lassen sich Namen noch einfacher merken.
  - *Es werden logische Gruppen (Domains) eingeführt, die dann wieder Untergruppen (Subdomains) enthalten können.*
- Mehrere IP-Adressen zu einem Namen sind möglich

DNS: Übersicht

- Bis zum Jahr 1984 wurden alle IP-Adressen und Rechnernamen in einer (zentralen) Datei (hosts.txt) verwaltet.
  - Zuständig für die Datei war das Network Information Center des Defense Data Networks.
  - Jeder im ARPAnet betriebene Host verwendete diese Datei zur Namensauflösung. Die Datei wurde dazu per ftp heruntergeladen.
- Problem: Dieser Ansatz skaliert nicht.
- Die Datei wurde durch eine verteilte „Datenbankanwendung“ ersetzt, den Domain Name Service oder auch Domain Name System (DNS).
- Das **Domain-Name-Service-Protokoll (DNS)** baut auf dem User-Datagram Protokoll (UDP) und der Transportschicht auf.
- Die „Well Known“ Portnummer des DNS-Protokolls ist die 53.
- Vorteile des Domain Name Service:
  - Übertragung von Host-Tabellen wird auf ein Mindestmaß reduziert.
  - Die Netzwerkverwaltung hat die Möglichkeit das Netz und seine Domain in autonome Verwaltungseinheiten (über Subnetting) aufzuteilen und die Verantwortung für die einzelnen Verwaltungseinheiten zu delegieren.
  - Relativ risikolose Änderung von Adressen und/oder Namen.

DNS Namen



- DNS-Namen sind hierarchisch (wie ein Baum) aufgebaut.
- Die Wurzel des Baums wird die Wurzel-Domäne (engl. root domain) genannt.
- Die Teilbäume werden **Domäne,** (engl. domain) genannt.
- Die Blätter des Baums sind i.R. dann die Rechnernamen (engl. host name). Kindknoten unter einer Domain müssen einen eindeutigen Namen haben.
- Die vollständigen Namen (FQDN) werden von unten nach oben gebildet. Für jede Baumkante wird ein Punkt (.) in den Namen gesetzt.
- Beispiel**: server01.inform.hs-hannover.de.**
- Der gesamt Baum ist der **DNS-Namensraum.**
- Die zweite Ebene im Baum wird **Top Level Domain** genannt, die Ebene darunter wird **Second Level Domain** genannt.

Aufbau von DNS Namen

- Um den Namen eines Hosts anzugeben, kann
  - der eigentlichen Host-Name, z. B. server01 oder
  - der voll-qualifizierte Domänenname (engl. fully qualified domain name) (FQDN) angegeben werden.
- Der FQDN wird gebildet, indem im DNS-Hierarchie-Baum von unten nach oben gelesen wird und jede Kante durch einen Punkt ersetzt wird. ***Beispiel: server01.inform.hs-hannover.de.***
- Die allgemeine Darstellung des FQDN sieht wie folgt aus:
  Rechnername.[Subdomain.]SecondLevelDomain.TopLevelDomain.
- FQDN müssen eindeutig sein.
- Kind-Namen innerhalb einer Domain müssen **eindeutig** sein.

- Erlaubte Zeichen sind in einem DNS-Namen fast alle Unicode-Zeichen.
- Weiterhin gibt es gibt zwei „Namens-Typen“:
  - **Absolute Namen** (vgl. absolute Pfadnamen in Dateisystemen) enden mit “.“ Punkt.
    (Beispiel: www.gmx.net.)
  - **Relative Namen** enden ohne Punkt. Die Software auf dem Client muss dann eine (oder mehrere) lokale Domains anhängen.
    (Beispiel: Name=ssh und lokale Domain=inform.hs-hannover.de. ergibt zusammengesetzt ssh.inform.hs-hannover.de.)

DNS Aufgaben und Funktionsweise

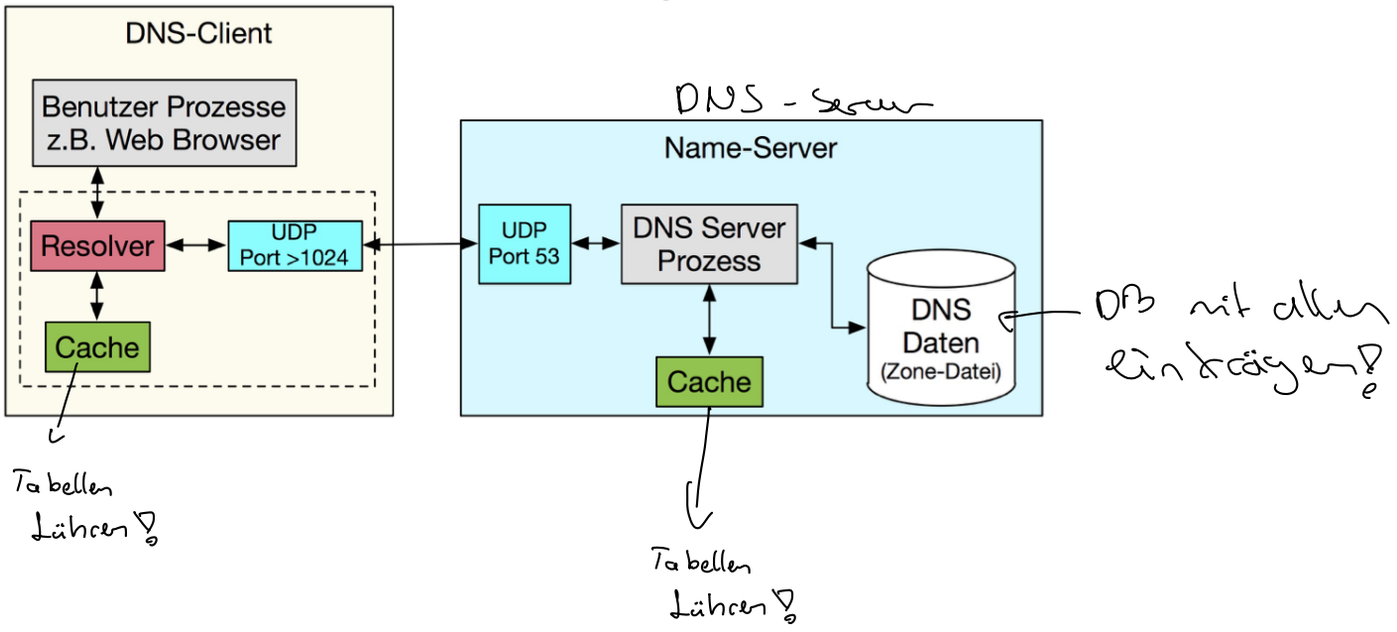
DNS Aufgaben

- Stelle die Zuordnung zwischen Namen und IP-Adressen zur Verfügung:
  - Abbildung von **Name** auf IP-Adresse (Forward[-Lookup] Zone)
  - Abbildung von **IP-Adresse** auf Name (Reverse[-Lookup] Zone)
- Dazu braucht ein DNS-Server (ab hier Name-Server (NS) genannt) Informationen über die Namen und IP-Adressen im gesamten Internet.
  - Datei /etc/hosts enthielt früher alle Namen und Adressen
  - Heute: verteilte „Datenbank“
- Aufgaben eines Name-Servers einer Domain X:
  - Kennt Namen/IP-Adressen aller Rechner der Domain X, für die der NS zuständig ist.
  - Kennt die IP-Adressen aller NS die für eine Sub-Domain von X zuständig sind.
  - Kennt die IP-Adressen der NS der Root-Domain.

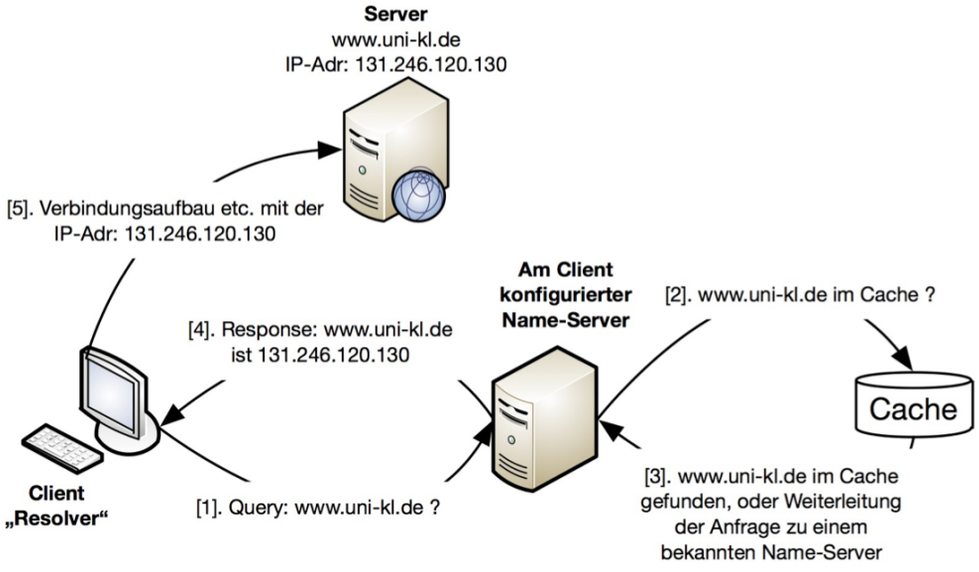
DNS Software Architektur

Client/Server Architektur mit:

- **Client** (Resolver, Stub-Resolver) ist für Anfragen zuständig.
- **Name-Server** wartet an UDP-Port 53 auf Anfragen und liefert Antworten.



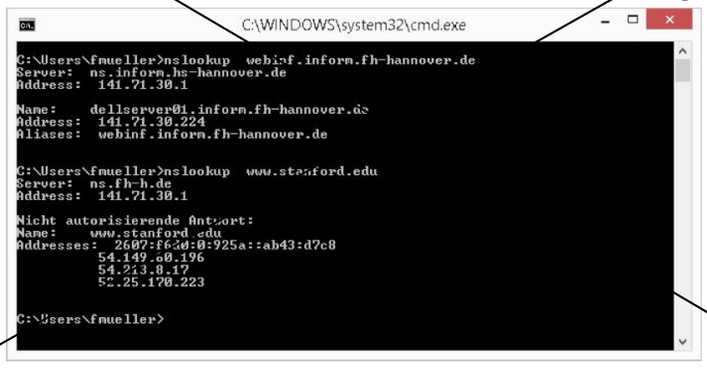
DNS-Anfrage (Prinzip/Transparenz/Rekursiv)



DNS-Anfragen (Resolver)

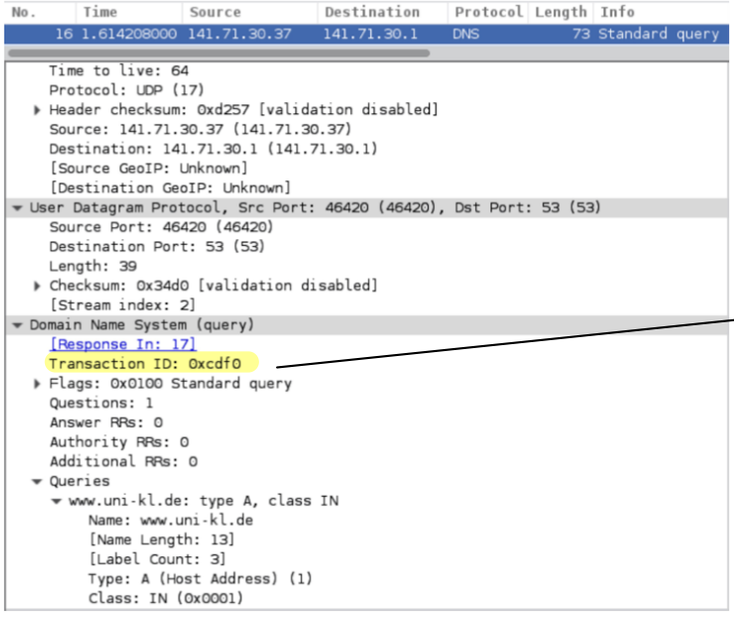
Kommando unter Unix und Windows heißt gleich: **nslookup**

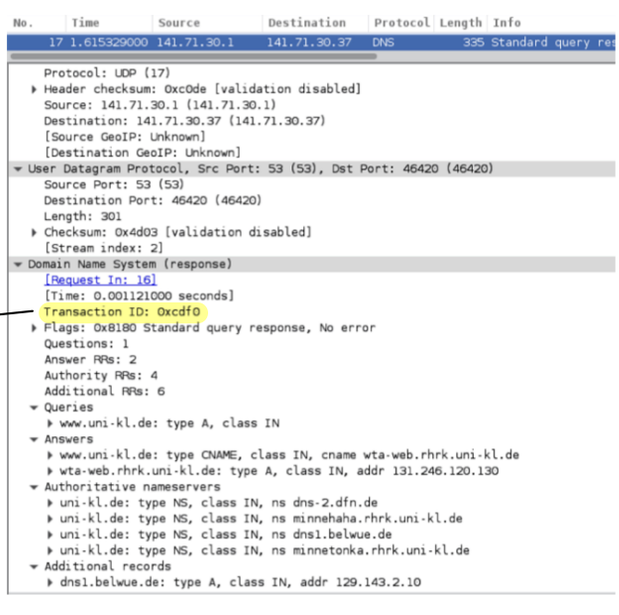
Weitere Kommandos unter Unix existieren, z. B.: host, dig, . . .



Ablauf einer DNS Anfrage

- Der Resolver prüft seinen lokalen Cache und/oder seine lokale **hosts** Datei.
- Falls **kein** Eintrag vorhanden, dann schickt der Resolver ein UDP-Anfrage-Paket an den konfigurierten Name-Server.
- Beim Resolver kommt ein UDP-Paket als Antwort an.
- In der Antwort stehen weitere Informationen!
- Die Antwort des Name-Servers wird im Cache des Resolvers gespeichert.





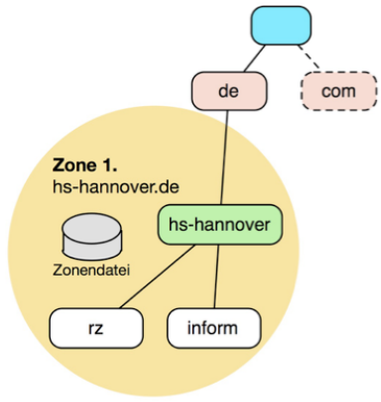
DNS-Domain und DNS-Zone

- Bei einer Domain wie z. B. hs-hannover.de kann ein eigener Name-Server eingesetzt werden, um die Namen und IP-Adressen der Hosts in dieser Domain zu verwalten.
- Die dafür notwendige Datenbank wird auf dem Name-Server in Form einer **Zonendatei** abgelegt.
- **DNS-Domain:**
  - Jeder Teilbaum im DNS-Namensraum ist eine Domain.
- **DNS-Zone:**
  - „Eine Zone ist die autorisierende Quelle für Informationen zu jedem DNS-Domänennamen, der in der Zone enthalten ist."
  - Ein Ausschnitt aus dem DNS-Namensraum für den ein einzelner Name-Server zuständig ist, d.h. der Name-Server kennt alle Rechner dieser Zone.

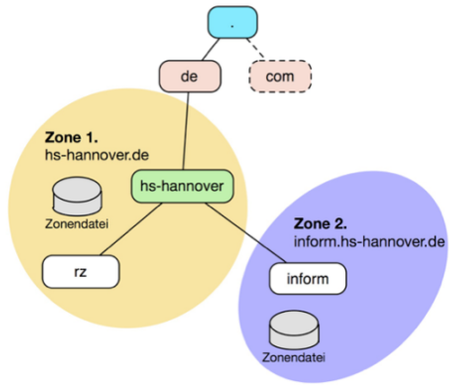
- Die Hochschule Hannover mit der Domain hs-hannover.de besteht aus weiteren Organisationseinheiten.
  - z. B.: Rechenzentrum, Betriebswirtschaft, Informatik, Wirtschafts-Informatik, etc.
- Diese Organisationseinheiten lassen sich in Form von Subdomains beschreiben.
  - Die Organisationseinheit der Abteilung Informatik z. B. als die Subdomain inform.hs-hannover.de
  - Die Organisationseinheit der Zentral-IT z. B. als die Subdomain rz.hs-hannover.de

Beispiel: DNS-Zone

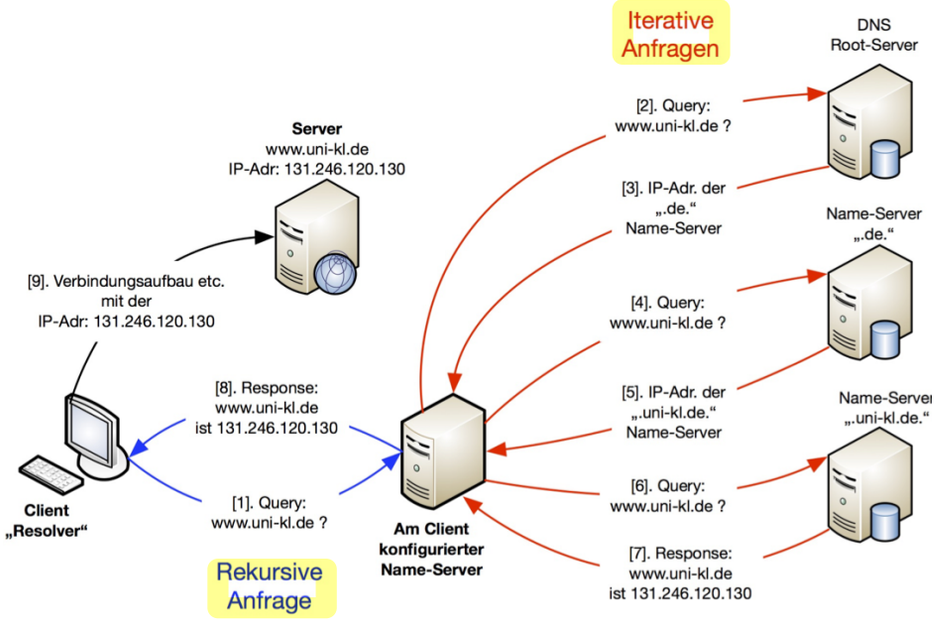
- **Stammdomain**: hs-hannover.de
- Untergeordnete **Subdomains**:
  - rz.hs-hannover.de inform.hs-hannover.de
- Die **Stammdomain** und alle ihre **Subdomains** werden in einer gemeinsamen **Zonendatei** von einem Name-Server verwaltet.
- **Vorteile:**
  - Wenig Hardware
  - Nur eine Zonendatei
- **Nachteile:**
  - Verwaltung falls Subdomains an anderem Ort? Performance eventuell unbefriedigend.



- Ausgliederung der **Subdomain** inform.hs-hannover.de in eine eigene Zone.
- Die **Subdomain** inform.hs-hannover.de wird in einer eigenen **Zonendatei** und von einem eigenen Name-Server verwaltet.
- Die **Zonendatei** der **Subdomain** inform.hs-hannover.de beinhaltet alle Hostnamen und IP-Adressen für die Domain inform.hs-hannover.de und die Adresse der übergeordneten Domain hs-hannover.de.



Zusammenarbeit von Name-Servern



Unterschiedliches Verhalten der DNS Server

Rekursives Verhalten (engl. recursive)

- Der Name-Server liefert entweder die komplette Antwort
- oder eine Fehlermeldung.
- Der Client-Resolver soll keinen unnötig komplexen Code enthalten.
- Daher wird in der Regel rekursiv vorgegangen
- Aus Client-Sicht transparent.

- **Beispiel**: Der am Client konfigurierte Name-Server auf der vorherigen Folie.

Iteratives Verhalten (engl. iterative)

- Der Name-Server liefert als Antwort einen Verweis (IP-Adressen) auf einen anderen Name-Server.
- Ein Name-Server soll möglichst wenig Arbeit haben.
- Root-NS antworten arbeiten daher iterativ

- **Beispiel**: Die anderen Name-Server auf der vorherigen Folie.

DNS Optimierungen und Sicherheitsprobleme

Optimierungen im DNS

1. Zwischenspeichern von neuen Adressen und Werten im Name-Server (engl. caching) oder auch im Resolver eines Clients.
2. Rückfragen nicht beim Root-Server starten, sondern bei einem Name-Server, der „näher“ am angefragten DNS-Namen liegt.
3. Zur Entlastung des Name-Servers einer Zone, einen zweiten Name-Server für diese Zone bereithalten.
   - ***Problem**: Konfigurationsdateien müssten dann auch zwei Mal existieren und konsistent gehalten werden.*
   - ***Lösung**: Betreibe Secondary NS, der seine Daten vom Primary NS bezieht.*

Dynamisches DNS

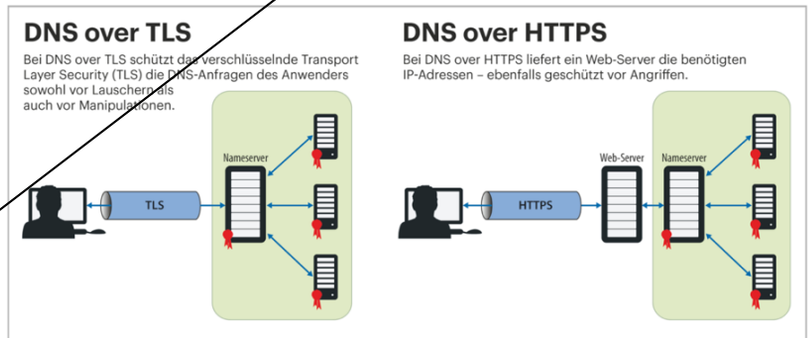
- **Herausforderung**: Ein Rechner hat eine IP-Adresse, die sich häufig ändert und sein DNS-Name soll immer auf die aktuelle IP-Adresse “verweisen“.
  - Laptop, das immer wieder in anderen Netzen ist.
  - Server zu Hause, der vom Internet Service Provider (ISP) jede Nacht eine neue IP-Adresse bekommt.
  - IPv6 Client, der privacy extensions benutzt.
- **Lösung**: Die Lebensdauer (TTL) eines DNS-Eintrags wird verkürzt und der Eintrag automatisch aktualisiert.
  - **Nachteil**: Eintrag schlecht im Cache haltbar.

Allg. Sicherheitsprobleme von Name-Servern

- **Ausfall eines Name-Servers:**
  - Installiere mehrere Name-Server für eine Zone. Einer der Server ist der Primary
  - NS, die anderen bekommen ihre Daten vom Primary NS geliefert.
- **DNS-Spoofing:**
  - Ein Rechner gibt vor einen anderen Namen zu haben, als er tatsächlich hat. Er „fälscht, bzw. täuscht“ (engl. to spoof) seine Identität.
- **Veränderung der Nachrichten bei der Übertragung:**
  - Alle Nachrichten sind unverschlüsselt und können von jedem abgehört werden. Außerdem können Nachrichten manipuliert werden, d.h. gefälschte Nachrichten können in den Cache des Resolvers oder eines NS gelangen (engl. cache poisoning).
- **Die Konsistenz** der Forward- und Reverse-Zone wird nicht erzwungen.

Aktuelle Ansätze zur Absicherung von DNS

DNS over HTTPS ist bpsw. inzwischen in Firefox eingebaut und kann manuell aktiviert werden.



Zusammenfassung

- Gründe für die Einführung von DNS-Namen?
- Verständnis des Namensraumes (Domain, Subdomain)
- Aufgaben eines Name-Servers (NS) und Zuständigkeiten (Zone)
- Unterschied zwischen einer Domain und einer Zone
- Arbeitsweise bei DNS-Anfragen (Rekursiv vs. Iterativ)
- Optimierungsmöglichkeiten
- Sicherheitsprobleme