Introduction to DNS (RFC 1034, RFC 1035) DNS vulnerabilities

Referat von Florian Oerterer

11.05.2004

Gliederung

- DNS im Schichtenmodell
- Einleitung
- Geschichte des DNS
- Funktionsweise
- Recource Records
- Message Format
- Beispiel einer DNS-Abfrage
- Sicherheitsaspekte des DNS
- Schwachstellen des DNS

DNS im OSI-Schichtenmodell

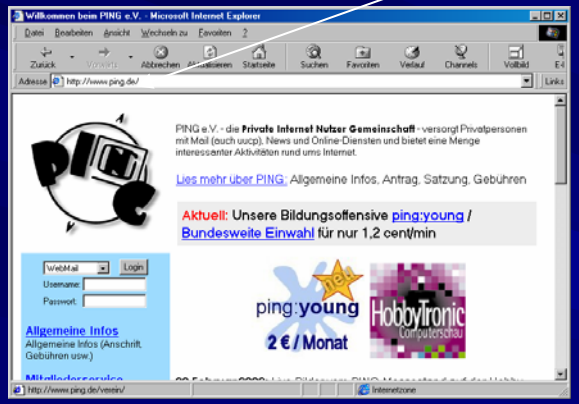
Einleitung (1/2)

- Der Mechanismus, der eine Rechner-Namenshierarchie für das Internet zur Verfügung stellt, wird DNS (Domain Name System) genannt.
- Umwandlung von ASCII-Netzadressen in numerische IP-Adressen
- Beispiel:

  www.unizh.ch wird zu 130.60.7.22

Einleitung (2/2)

Eingabe im Browser: **`www.ping.de`**



Der Browser fragt einen DNS-Server nach der IP-Adresse von www.ping.de

Der DNS-Server antwortet mit der Adresse 62.72.90.2

DNS-Server

Der Browser kontaktiert 62.72.90.2 und fragt nach der gewünschten Seite

62.72.90.2 schickt die Daten zurück und der Browser stellt sie dar

62.72.90.2 (www.ping.de)

Geschichte des DNS

- Frühere Systeme unterstützen nur Punkt zu Punkt Verbindungen zwischen Rechnern anhand von Hardware Adressen
- Nächster Schritt: Vergabe von Namen anstelle der Hardware-Adresse
- Im Folgenden:
  - IP-Adressen = low level names
  - Maschienen-Namen = high level names
- 1980 – gab es einige Dutzend Rechner
- 1986 – gab es 3100 offizielle Namen, 6500 Alias-Namen
- 1990 – gab es 6400 registrierte Namen beim NIC(Network Information Center) , aber obsolete, da ca. 137.000 Namen im Internet Domain Name System

Domänen

- Hierarchische Aufteilung des Internets
- Es entsteht ein Baum von Domänen
  - Jede Domäne ist in Teildomänen unterteilt
  - Die Blätter sind Domänen, welche schlussendlich Hosts enthalten

Beispiel: www.informatik.fh-nuernberg.de

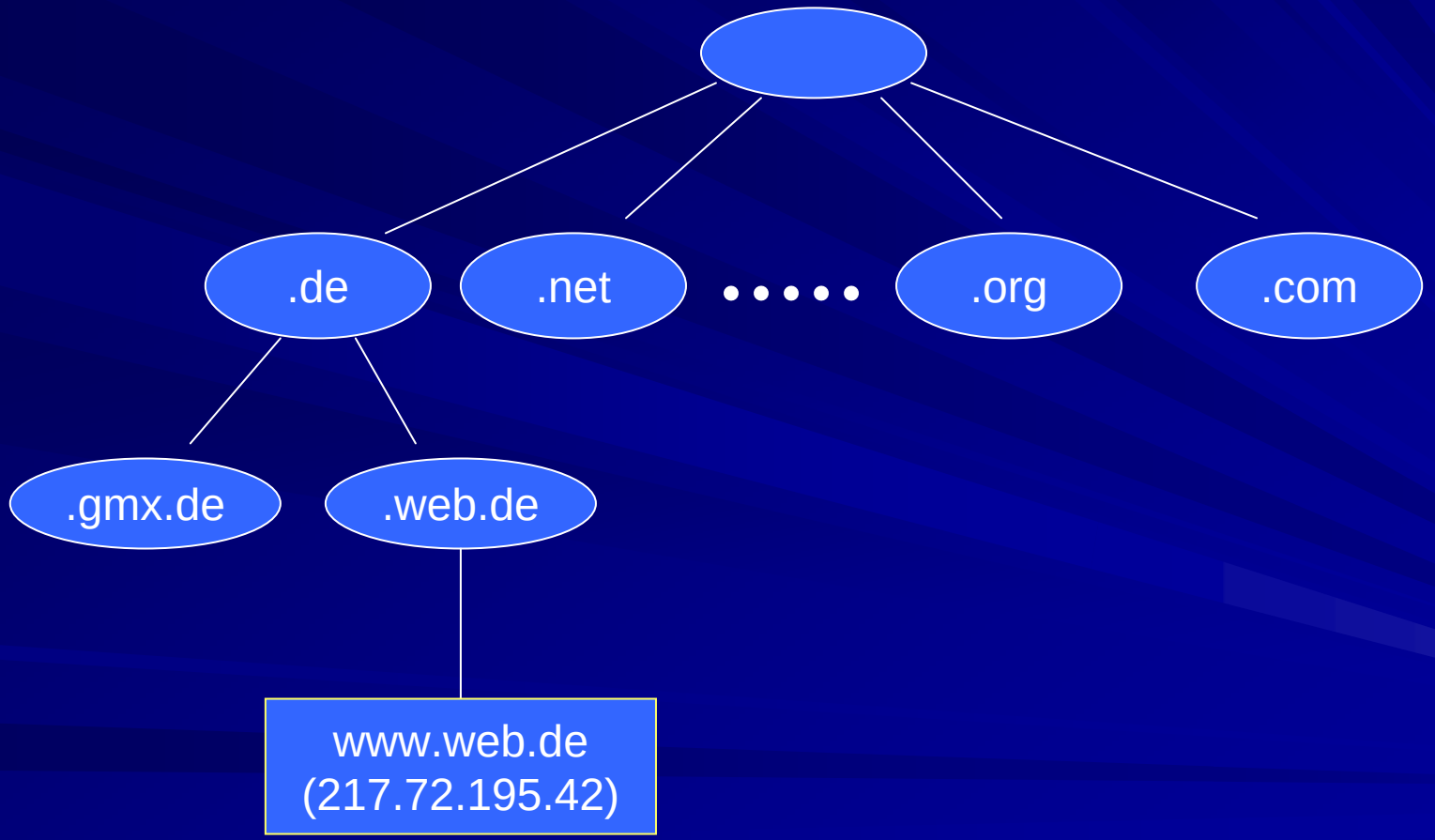
Lowest-Level-Domain.Subdomain.Subdomain.Top-Level-Domain

- =>Kein Server enthält alle Informationen des DNS!!

Namenshierarchie (1/4)

- Früher flacher Namensraum
  - Potentiell hohe Konflikte
  - Administrativer Aufwand an einer Stelle sehr hoch
- Alternative: Hierarchischer Namensraum
- Internet-Autorität vergibt einen eindeutigen Domain-Suffix.
  Beispiel Deutschland :
  Oberster Server: DeNIC
  Domain : .de
- Jede Domain repräsentiert einen Teilbaum.
- Anzahl oder Struktur der Subdomains beliebig.
- Der Namensraum muss nicht geographisch strukturiert sein.
- Einschränkungen:
  - Eine Adresse muss mindestens aus 3 Teilen bestehen.
  - Jedem Teil ist eine Maximallänge von 63 Zeichen zugeordnet.
  - Jeder Adresse ist eine Maximallänge von 255 Zeichen zugeordnet

Namenshierarchie (2/4)



Namenshierarchie (3/4)

- Domain / Subdomain Naming Conventions
  - Theoretisch erlaubt *DNS standard* beliebige Werte für Labels
  - Jedoch anfangs Beschränkung durch IAB (Internet Architecture Board) für Top Level Domain auf:

Namenshierarchie (4/4)

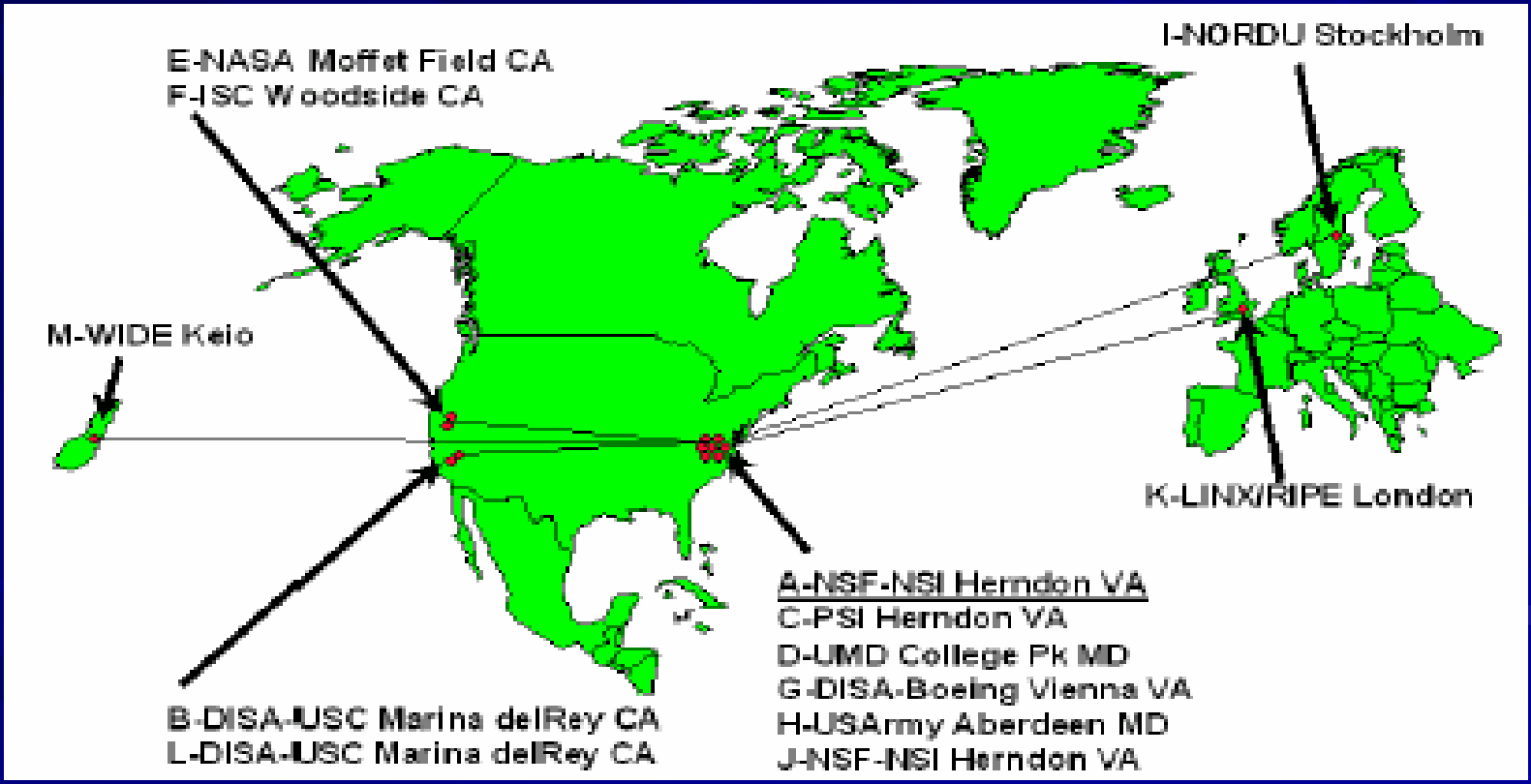
Domain Name	Bedeutung
COM	Commerzielle Organisationen
EDU	Erziehung, Hochschulen, etc.
GOV	Regierungsinstitutionen
MIL	Militärische Einrichtungen
NET	Netzwerk-Support-Zentren
ORG	Andere Organisationen
ARPA	ehemals ARPANET Domains (Obsolete)
INT	Internationale Organisationen
country code	Länderbezeichnung für jedes Land

Serverhierarchie (1/3)

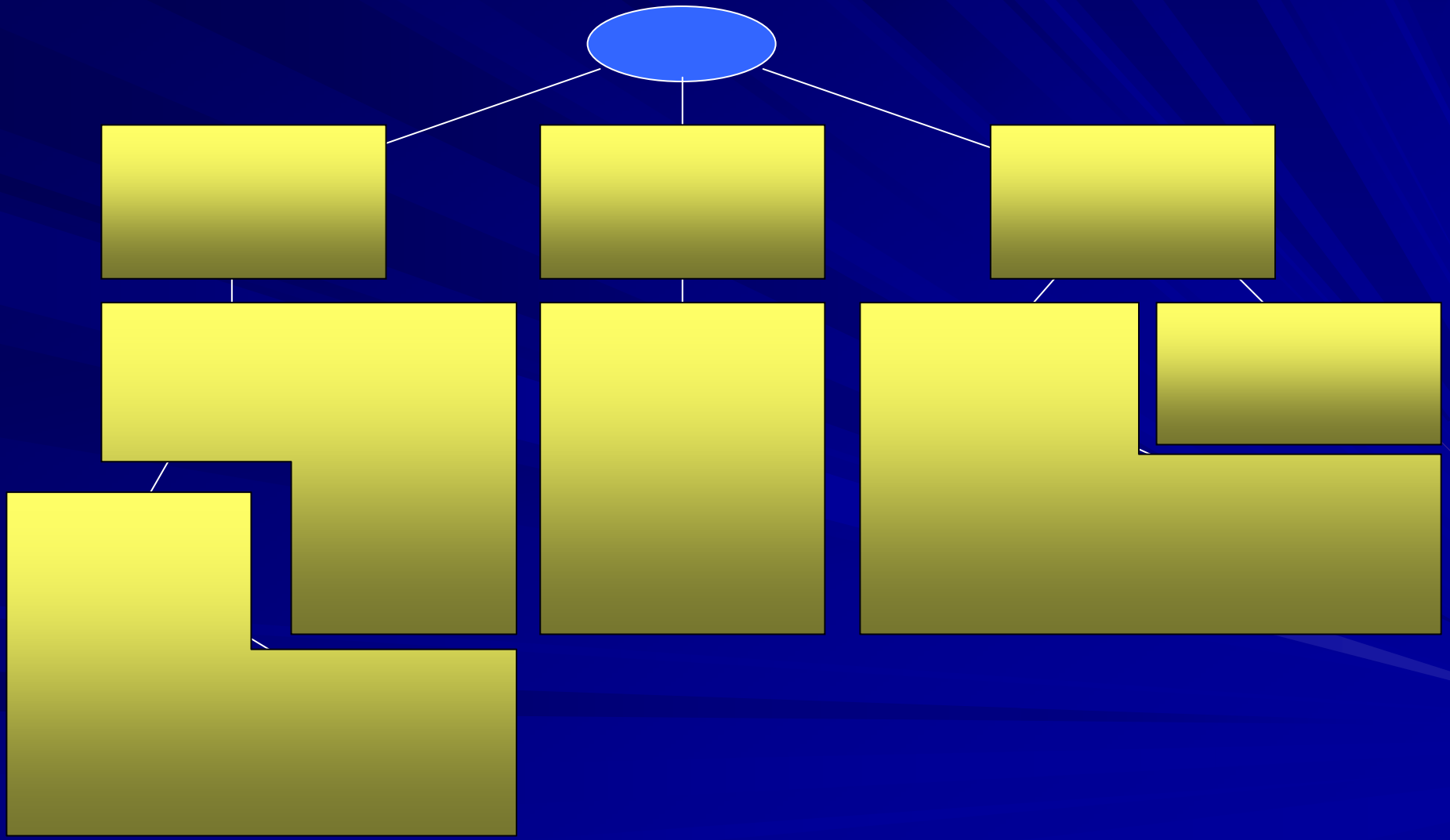
- DNS-Server sind, wie der Namensraum, hierarchisch organisiert.
- Der Server muss nicht alle Namen kennen, aber er muss andere Server kennen, die für diese Subdomains verantwortlich sind.
- Eine bestimmte Ebene der Namenshierarchie kann unter verschiedene Server aufgeteilt werden.
- Der “Autoritäts“-DNS-Server verwaltet die Übersetzungstabelle.
- Ein verantwortlicher Server hat die Übersetzungstabelle, oder er kennt einen Server, der sie hat.

Serverhierarchie (2/3)

- Derzeit 13 Root-Nameserver weltweit



Serverhierarchie (3/3)



Die Inverse Abfrage

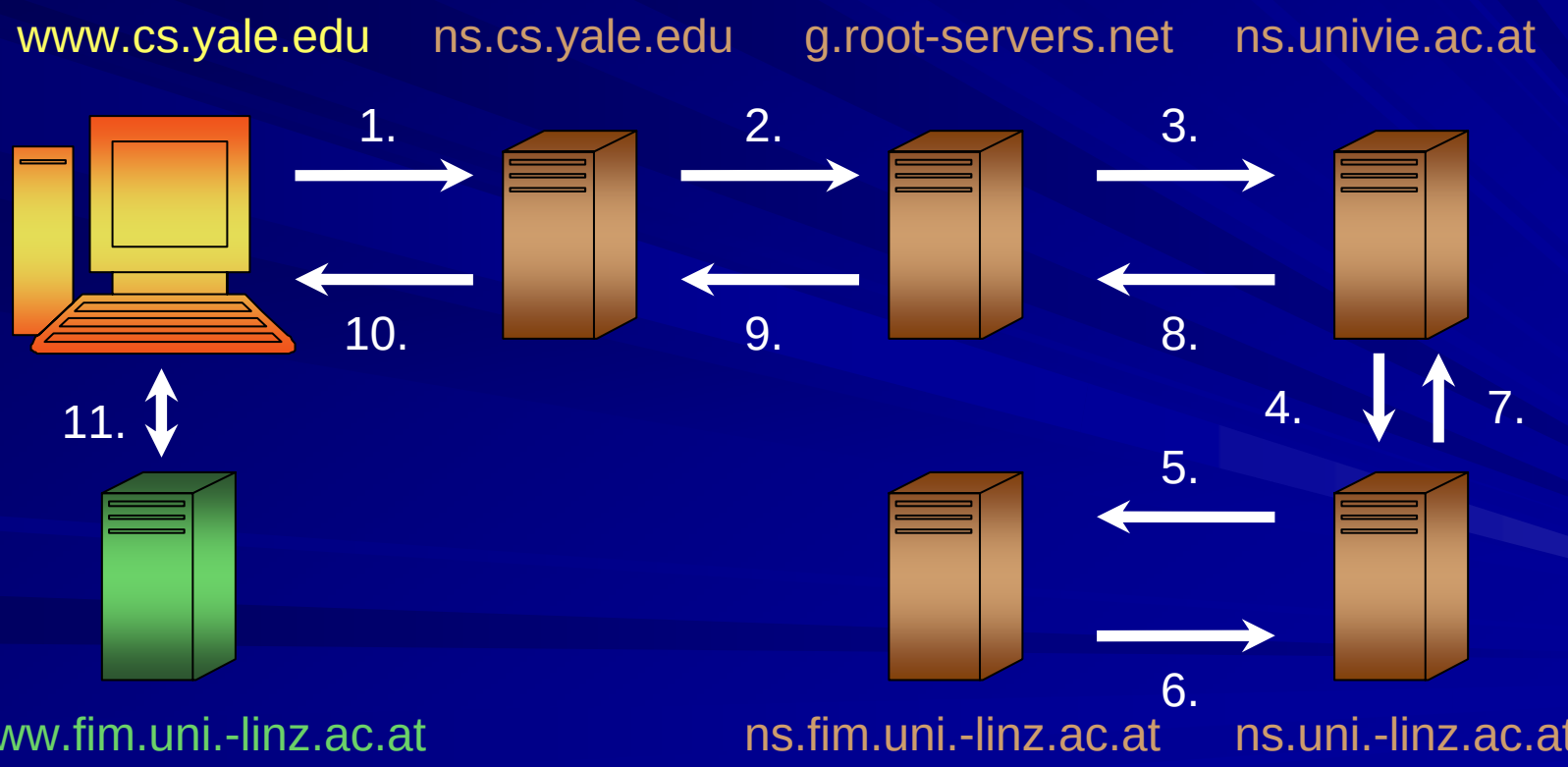
- Die spezielle Domain in-addr.arpa
- Dient zur Auflösung von IP-Adressen in Namen (inverses DNS)
- Jedes Oktett einer IP bildet eine Subdomain
- Bsp.: 141.75.149.10

  => 10.149.75.141.in-addr.arpa

  => www.informatik.fh-nuernberg.de

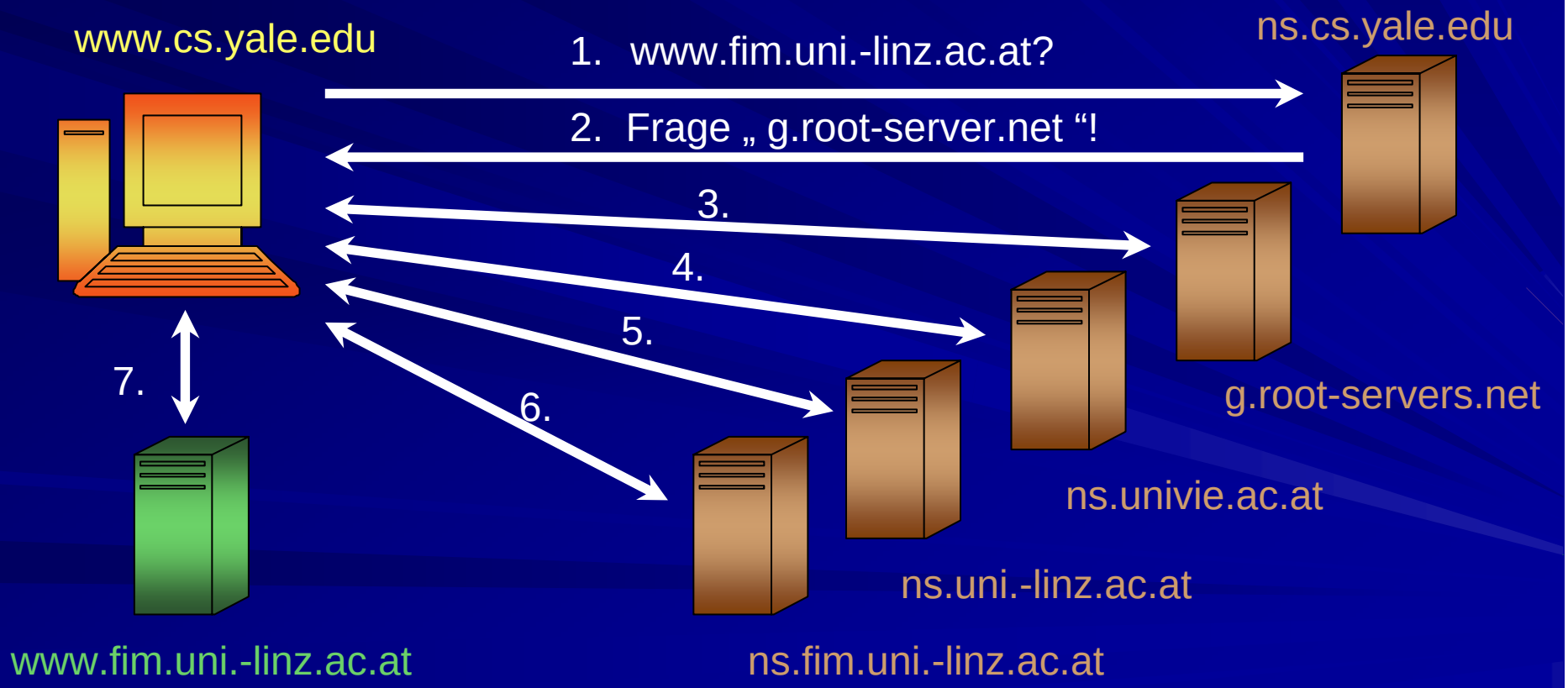
Die Rekursive Abfrage

- Die Anfrage wird von Nameserver zu Nameserver weitergeleitet, bis der autoritive Server gefunden ist



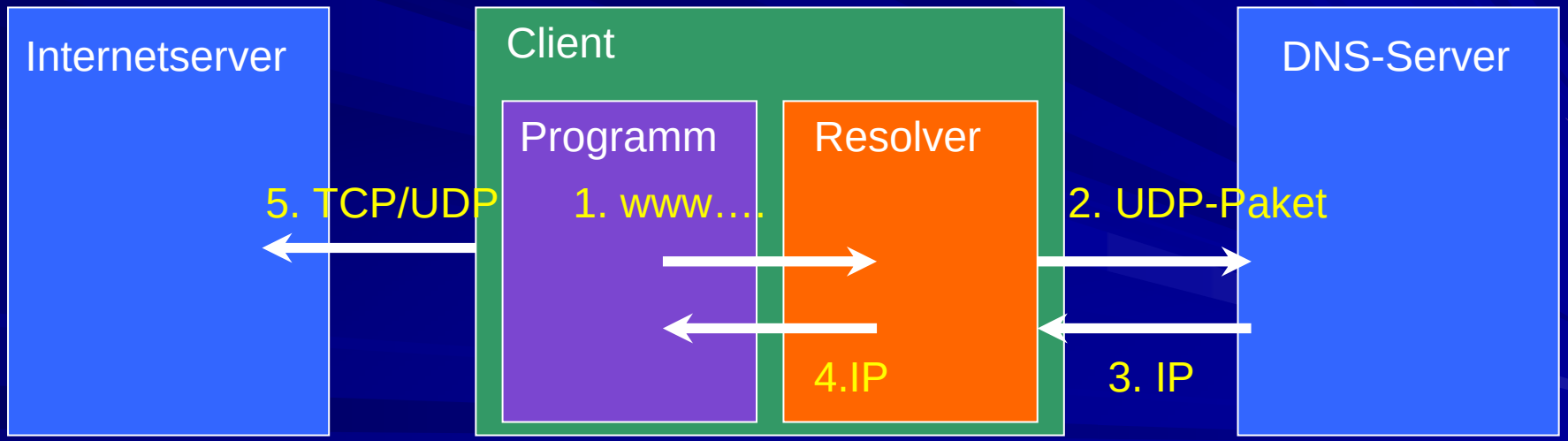
Die Iterative Abfrage

- Der Nameserver erhält auf seine Frage jeweils eine Liste mit Namenservern zurück, die er als nächste fragen soll.



Der Resolver

- Definition: Bibliothek von Routinen, welche Anfragen über Rechner an den name-Server formulieren kann



DNS - Caching

- Namensserver nutzt Caching zur Kostenersparnis und Effizienzsteigerung
- Jeder Server hat einen Cache für kürzlich benutzte Namen und Informationen darüber, wo sie stammen bzw. wann sie veralten
- Name Server geben *nonauthoritative answers* und den Namen des Servers zurück, von dem diese Informationen stammen
- Es folgt:
  - Schnell, aber evtl. „out of date“
  - Effizient, wenn nonauthoritative answer reicht
  - Richtig, dann Kontakt zu autorisiertem DNS notwendig
  - TTL Feld gibt jeweils Aufschluss, wann Info „out of date“

Ressource Records (1/5)

Format von Resource Einträgen

- Jeder Answer-, Authority- und Additional Info Section besteht aus sogenannten Recource-Records:

Resource Domain Name ……	
Type	Class
Time To Live	Resource Data Length
Resource Data ……	

- Type gibt den Type der Frage an
- Class gibt die zugehörige Klasse des Objektes an
- Time_to_live: Angabe, wie stabil der Eintrag ist
- Recource Data Length gibt die Länge des Resource Data Feldes an
- Resource Data enthält die gewünschten Infos

Ressource Records (2/5)

Typ des Records:

Type	Bedeutung	Wert
SOA	Start of Authority	Parameter für die betreffende Zone
A	IP-Adresse eines Hosts	32-Bit Ganzzahl
MX	Mail Exchanger	Priorität, in der die Domäne E-Mail annimmt
NS	Name Server	Name eines Servers der betreffenden Domäne
CNAME	Canonical Name	Übersetzt einen Alias der Domain in den echten Namen
PTR	Pointer	Alias für eine IP-Adresse
HINFO	Host description	CPU und Betriebssystem in ASCII
WKS	Well Known Service	List der Dienste, die der Rechner anbietet
TXT	Text	Uninterpretierter ASCII-Text

Ressource Records (3/5)

<host> [<ttl>] [<class>] **A** <address>
<nickname> [<ttl>] [<class>] **CNAME** <host>
<host> [<ttl>] [<class>] **HINFO** <hardware><software>
<host> [<ttl>] [<class>] **WKS** <address><protocol><services>
<name> [<ttl>] [<class>] **MX** <preference><host>
<domain> [<ttl>] [<class>] **NS** <server>

<spec.name> [<ttl>] [<class>] **PTR** <name>
<name> [<ttl>] [<class>] **SOA** <origin><person>
(<serial><refresh><retry><expire><minimum>)

Ressource Records (4/5)

- Beispiel: Teil einer Datenbank mit 7 Ressourcendatensätzen

;Authoritative data for cs.vu.nl

cs.vu.nl.	86400	IN	SOA	star boss (952771, 7200, 7200, 2419200, 86400)
cs.vu.nl.	86400	IN	TXT	„Faculteit Wiskunde en Informatica.“
cs.vu.nl.	86400	IN	TXT	„Vrije Universiteit Amsterdam.“
cs.vu.nl.	86400	IN	MX	1 zephyr.cs.vu.nl.
cs.vu.nl.	86400	IN	MX	2 top.cs.vu.nl.

flits.cs.vu.nl.	86400	IN	HINFO	Sun Unix
flits.cs.vu.nl.	86400	IN	A	130.37.16.112
flits.cs.vu.nl.	86400	IN	A	192.31.231.165
flits.cs.vu.nl.	86400	IN	MX	1 flits.cs.vu.nl.
flits.cs.vu.nl.	86400	IN	MX	2 zephyr.cs.vu.nl.
flits.cs.vu.nl.	86400	IN	MX	3 top.cs.vu.nl.
www.vs.vu.nl.	86400	IN	CNAME	star.cs.vu.nl
ftp.cs.vu.nl.	86400	IN	CNAME	zephyr.cs.vu.nl

Ressource Records (5/5)

rowboat	IN	A	130.37.56.201
	IN	MX	1 rowboat
	IN	MX	2 zephyr
	IN	HINFO	Sun Unix
little-sister	IN	A	130.37.62.23
	IN	HINFO	Mac MacOS
laserjet	IN	A	192.31.231.216
	IN	HINFO	„HP Laserjet IIISi“ Proprietary

DNS Message (1/4) - Format

0	15 16	31
IDENTIFICATION	PARAMETER	
No. OF QUESTIONS	No. OF ANSWERS	
No. OF AUTHORITY	No. OF ADDITIONAL	
QUESTION SECTION		
……		
ANSWERS SECTION		
……		
AUTHORITY SECTION		
……		
ADD. INFO SECTION		
……		

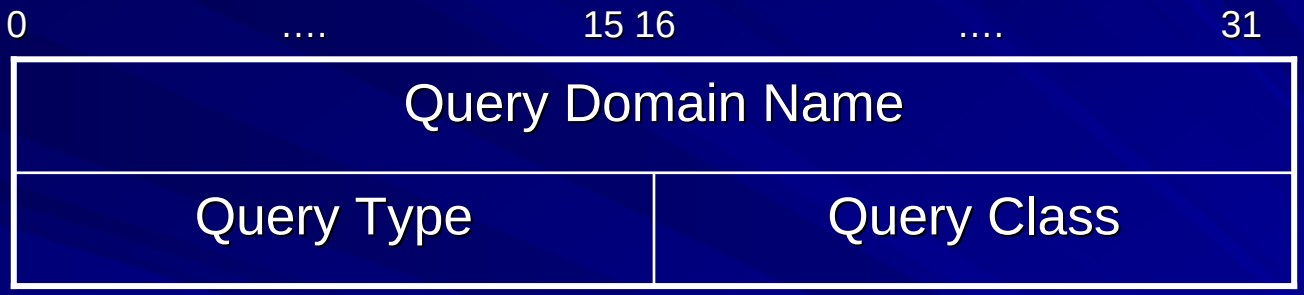
Identification	Eindeutige Kennung zur Abbildung von Fragen und Antworten
PARAMETER	s.u.
No. OF …	Anzahl der Entries in dem Bereich
Question Section	Client trägt Fragen in die Question Section ein, Server füllt Answer, Authority- und Additional Info Section.
Answer Question	
Authority Section	
Add. Info Section	

DNS Message (2/4) - Parameter Bits

Bits des Parameter-Feldes	Bedeutung
0	Operation: 0=Query, 1=Response
1 – 4	Query Type: 0=Standart, 1=Inverse, 2 und 3 obsolete
5	Gesetzt, wenn Antwort autorisiert
6	Gesetzt, wenn Nachricht abgeschnitten
7	Gesetzt, wenn Rekursion notwendig
8	Gesetzt, wenn Rekursion verfügbar
9 – 11	Reserviert
12 – 15	Response Type: 0=No Error, 1=Format Error, 2=Server Fehler, 3=Name unbekannt

DNS Message (3/4)

Format von Query Einträge



- Query Type gibt den Type der Frage an, ob

z. B. nach einer Adresse für einen

Rechnernamen gesucht wird oder nach

einer Mailbox, einer Host Info etc.

- Query Class gibt an, zu welcher Klasse das gesuchte Objekt gehört. Hier ist für die Klasse *IN* wichtig

DNS Message Format (4/4)

Darstellung von Domain Names in Nachrichten

- Jeder Domain Name wird als eine Folge von Labels dargestellt.
- Jedes Label beginnt mit einem Längenbyte.
- Ein Byte 0 beendet den Namen
- Längenfeld wird teilweise auch als Pointer-Feld benutzt, um Speicherplatz zu sparen (mehrfaches Auftreten der gleichen Domain). Es verzweigt dann an die Stelle, wo der Rest der Informationen steht.

DNS Abfragen (BIND)

- BIND (Berkley Internet Name Daemon)
- BS=UNIX
- Populärste Implementierung des DNS Protokolls
- Besteht aus:
  - Resolver
  - Server Routinen
  - Tools (wie NSLOOKUP)

NSLOOKUP (Beispiel1)

zdv104@zam108>**nslookup**
Default Server: zam049.zam.kfa-juelich.de
Address: 134.94.80.3
> **set query=NS**
> **fh-aachen.de**
Server: zam049.zam.kfa-juelich.de
Adress: 134.94.80.3
Non-authoritative answer:
fh-aachen.de nameserver = nets1.rz.rwth-aachen.DE
fh-aachen.de nameserver = lroda0.bau.FH-Aachen.DE
fh-aachen.de nameserver = iris0.dvz.FH-Aachen.DE
Authoritative answers can be found from:
FH-Aachen.de nameserver = nets1.rz.rwth-aachen.DE
FH-Aachen.de nameserver = lroda0.bau.FH-Aachen.DE
FH-Aachen.de nameserver = iris0.dvz.FH-Aachen.DE
nets1.rz.rwth-aachen.DE inet address = 137.226.144.3
lroda0.bau.FH-Aachen.DE inet address = 149.201.60.60
Iris0.DVZ.FH-Aachen.de inet address = 149.201.10.30
Iris0.DVZ.FH-Aachen.de inet address = 149.201.10.29

NSLOOKUP (Beispiel2)

zdv104@zam108>**nslookup**

Default Server: zam049.zam.kfa-juelich.de

Address: 134.94.80.3

> **set query=MX**

> **ippnv2.ipp.kfa-juelich.de**

Server: zam049.zam.kfa-juelich.de

Address: 134.94.80.3

ippnv2.ipp.kfa-juelich.de

    preference = 100

    mail exchanger = ipp064.ipp.kfa-juelich.de

NSLOOKUP (Beispiel3)

zdv104@zam108>**nslookup**
Default Server: zam049.zam.kfa-juelich.de
Address: 134.94.80.3
>**set query=ANY**
>**aix.sp.kfa-juelich.de**.
Server: zam049.zam.kfa-juelich.de
Address: 134.94.80.3
aix.sp.kfa-juelich.de inet address=134.94.24.2
aix.sp.kfa-juelich.de preference 10,
mail exchanger = zam225.sp.kfa.juelich.de
>**aix.sp.kfa-juelich.de**.
Server: zam049.zam.kfa-juelich.de
Address: 134.94.80.3
aix.sp.kfa-juelich.de inet address=134.94.24.3
aix.sp.kfa-juelich.de preference 10,
mail exchanger = zam225.sp.kfa.juelich.de

Sicherheitsaspekte des DNS

Warum ist die Sicherheit in DNS-Servern besonders wichtig?

- Erreichbarkeit
  DNS arbeitet wie ein „Telefonbuch“.
  Wird eine Anfrage eines Browsers zu einer falschen IP-Adresse geleitet, kann dieses zur Verwirrung und zum Image-Schaden des Site-Inhabers führen.

- Vertraulichkeit
  Wird ein Anwender mittels gefälschter IP-Adresse auf eine andere Seite geführt, kann es zum Verlust sensitiver Daten oder Passwörter kommen.

Hackerangriffe (1/4)

- Zone Transfers
  - Primary Name Server gibt seine gesamten Daten an einen Hilfsserver weiter
  - Dient zur Entlastung des Primary Servers
  - Hacker kann sich als Hilfsserver ausgeben
- Read buffer overflow in den Resolver libraries
  - Für Antworten muss in der „DNS stub resolver library“ Speicherplatz reserviert werden (weniger als 64K)
  - Bei zu langen Antworten wird abgeschnitten
  - Dabei kann es evtl. zu abstürzen kommen

Hackerangriffe (2/4)

- Cache Poisoning/DNS-Spoofing
  - Ein Hacker besitzt einen Authority-Zugriff zu einer Zone
  - Hacker startet rekursive DNS-Anfrage nach einem Host
  - Zielserver fragt den Server des Hackers an
  - Hacker schickt falsche oder unnütze Informationen an Zielserver
  - Diese werden im Cache des Zielservers gespeichert
- Masquerading
  - Angreifer gibt sich als bestimmten Server aus
  - Der Cache des Angreifers enthält eine falsche IP
  - Benutzer wird auf eine falsche Website geführt

Hackerangriffe (3/4)

- DoS: Denial of Service
  - Der Zugriff soll generell verhindert werden
  - Hierbei wird wieder die Cache Poisoning – Methode verwendet
  - Folgende zwei Auswirkungen sollen erreicht werden
    - Absturz des Servers: Bsp.: falsch formatierte DNS-Pakete und der Read Buffer Overflow
    - Belegung der Bandbreite: Server wird durch Flut von Daten überlastet, sodass er externe Anfragen nicht mehr beantworten kann

Hackerangriffe (4/4)

- Reaktion auf falsch formatierte DNS-Packete
  - In einer DNS-Anfrage/Antwort können Labels mehrmals vorkommen
  - Laut RFC 1035 werden hierfür Pointer eingesetzt
  - Den Labels wird folgende **Längeninformation** vorangestellt (8 Bit):

Label	Label	Label	Label	Label	Label	PTR	PTR

  - Hacker ändern nun die Pointer, so dass der ganze Speicher belegt wird oder Endlosschleifen entstehen

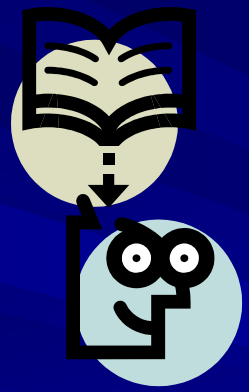
Maßnahmen

- Physikalischen Zugriff der DNS-Server mit Passwörtern sichern
- Regelmäßige Überwachung der Server
- Korrekte Zonenaufteilung der Nameserver
- Mindestens zwei Nameserver verwenden
  - Einer für rekursive Anfragen
  - Einer für DNS-Anfragen über seine Zone
- Eine möglichst aktuelle Version von Bind installieren
- Nameserver wird über Firewall geschützt

Quellen

- Internet
- Computernetze (Andrew S. Tannenbaum)
- TCP/IP-Grundlagen (Gerhard Lienemann)
- RFC 1034: CONCEPTS AND FACILITIES
- RFC 1035: IMPLEMENTATION AND SPECIFIKATION

Danke für die Aufmerksamkeit!



Fragen?