# The DNS they are changin'

## GUUG Frühjahrsfachgespräch 2026

Carsten Strotmann, sys4 AG

CREATED: 2026-03-06 FRI 06:24

# The DNS they are changin'

# Wer

- Carsten Strotmann
- DNS Nerd seit 1998
- Meine Themen: DNS/DNSSEC, DHCP, IPv6, DANE, Unix/BSD/Linux

# Worüber

- Das DNS Protokoll ist alt (1983)
- Das Internet und die Anwendungen verändern sich
- Die Anforderungen an das DNS wachsen
- Daher ändert sich auch das DNS
    - In diesem Vortrag werden interessante DNS-Änderungen der letzten Jahre vorgestellt

# Warum

- Einige der neuen DNS-Funktionen sind sinnvoll und hilfreich
    - auch wenn nicht alle neuen Funktionen für jeden umsetzbar sind

- Viele Netze in Firmen und Organisationen sind technisch in den 1990er Jahren stehengeblieben
- Inkrementelle Aktualisierung von Netzwerkfunktionen ist einfacher als ein "Big-Bang" nach Jahren der Vernachlässigung

# Über den IETF Standardisierungs-Prozess

# I-D und RFC

- **Jeder** kann in der IETF einen neuen Standard vorschlagen
  - Es wird ein Internet-Draft (I-D) geschrieben
  - Ein Draft **kann** als Arbeitsdokument einer IETF-Arbeitsgruppe (6MAN, DNSOP, ...) angenommen werden ...
  - ... oder als "individuelles" Dokument durch den RFC Prozess gehen
  - Alle I-D haben ein Ablaufdatum ... viele I-D erreichen nie RFC Status

# RFC Kategorien

- Nicht jeder RFC ist ein "Standard"!
- RFC-Kategorien:
  - *Standard* - dies sind die **echten** Standards
  - *Proposed Standard* - Protokolle werden aktiv eingesetzt und könnten einmal ein Standard werden
  - *Best Current Practice* - Empfehlungen zum Einsatz von Internet Protokollen
  - *Informational* - nur zur Information, keinerlei Standard, manchmal RFC-Parodien (April-Scherz-RFC)
  - *Experimental* - hier werden neue Protokolle getestet
  - *Historic* - veraltete Protokolle/Standards

# RFC 9460 - Die HTTPS/SVCB Records

# HTTP/3

- HTTP/3 (RFC 9114 - Juni 2022 - Proposed Standard) ist das aktuelle HTTP-Protokoll im Internet
- Es löst die älteren Protokolle HTTP/1.1 und HTTP/2 ab
- HTTP/3 wurde unter dem Namen *QUIC* bei Google entwickelt und dann in der IETF angepasst und standardisiert
- HTTP/3 benutzt UDP/443 statt TCP/443
- HTTP/3 gibt es nur in der Variante mit Verschlüsselung, daher sollte man es eher HTTPS/3 nennen

# Das Dilemma der Web-Browser

- Traditionell finden Web-Browser einen Web-Server über DNS-Auflösung des Server-Namens in AAAA (IPv6) oder A (IPv4) Adressen-Records
- Diese Record geben dem Browser keine Info, ob der Dienst über TCP/443 (HTTP < 3) oder UDP/443 (HTTP >= 3) erreichbar ist
    - Browser versuchen UDP/443, warten auf Timeout, dann Fallback auf TCP/443
    - nicht toll, Webseiten laden langsam

# Die HTTPS/SVCB-Records

- Die neuen DNS-Records **SVCB** (Type 64) und **HTTPS** (Type 65) bieten neben IP-Adressen zusätzliche Informationen für Netzwekdienste, wie das Protokoll für HTTPS-Verbindungen
  - Diese Records sind in RFC 9460 Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs) (November 2023, Proposed Standard) beschrieben
  - **SVCB** ist die generische Variante, und **HTTPS** ist die spezielle Version für HTTPS-Verbindungen

# Vorteile der HTTPS/SVCB-Records (gegenüber A/AAAA-Records)

- Definition von mehreren Server-Endpunkten (automatisches Failover von Diensten)
- IPv6- und IPv4-Adressen des Dienstes
- Information über die Protokoll-Version (HTTP1.1, HTTP/2 oder HTTP/3)
- Information über einen empfohlenen DNS-over-HTTPS (DoH) Resolver für eine Domain (Adaptive DoH Resolver Discovery)

# Vorteile der HTTPS/SVCB-Records (gegenüber A/AAAA-Records)

- Diese Records erlauben Alias-Namen für ganze Domains innerhalb einer Zone, eine oft nachgefragte Funktion welche mit `CNAME`- oder `DNAME`-Records nicht möglich sind (Beispiel: `firma.de` ist ein Alias für `firma.com`
- `SVCB` kann als eine Weiterentwicklung des älteren `SRV`-Record gesehen werden

# HTTPS/SVCB Verbeitung

- Die neuen HTTPS/SVCB-Records werden seit Sommer 2020 aktiv genutzt (Apple iOS 14 und macOS 11 waren die ersten Systeme)
    - Schon im Herbst 2020 war das HTTPS-Record an Platz drei der meist angefragten DNS-Records bei einem großem deutschen ISP
    - Heute benutzen alle modernen Browser und Betriebssysteme den HTTPS-Record
    - Wird kein HTTPS-Record gefunden (NODATA/NXRRSET), so gibt es ein Fallback auf AAAA- (langsam) und dann auf A-Record (noch langsamer)

# Beispiel des SVCB/HTTPS-Alias-Modus

- Beispiel SVCB im *Alias* Modus (für den `ldap` Dienst auf Port `389`)

```
_389._ldap.example.com. 3600 IN SVCB 0 ldap-server.exa
```

- Beispiel HTTPS *Alias* Modus

```
example.com. 3600 IN HTTPS 0 svc.example.net.
```

# SVCB/HTTPS im Service Modus

```
_port._service.ownername.   TTL  IN SVCB  <prio> <target> <k
```

- Das `prio` Feld funktioniert wie die Priorität in SRV- oder MX-Records (niedriger Wert = höhere Priorität)
- Das Feld `target` enthält den Hostnamen des Servers, welcher diesen Dienst bereitstellt. Der Hostname `.` bedeutet: *"benutze den Domain-Namen des SVCB/HTTPS-Record als Hostnamen für diesen Dienst"*

# HTTPS/SVCB Service Mode Parameter

- `mandatory` - Liste der obligatorischen Parameter
- `alpn` - Unterstützte Protokollvarianten
- `no-default-alpn` - Keine "Default" Protokollvariante
- `port` - Port für den Dienst auf dem Server
- `ipv6hint` - IPv6 Adressen (Hint)
- `ipv4hint` - IPv4 Adressen (Hint)
- `echconfig` - "Encrypted Client Hello" (TLS) Schlüssel

# Beispiel SVCB/HTTPS Service Modus

- HTTPS in *Service*-Modus (erster HTTPS-Record für *Alias*-Modus)

```
example.com.       7200  IN HTTPS 0 svc.example.net.
svc.example.net.   7200  IN CNAME svc2.example.net.
svc2.example.net. 7200  IN HTTPS 1 . port=8002 ipv6hin
svc2.example.net. 300   IN A     192.0.2.2
svc2.example.net. 300   IN AAAA  2001:db8::2
```

# HTTPS-Records ohne HTTP/3?

- Sollte man HTTPS-Records in der eigenen DNS-Zone einsetzen, auch wenn der eigene Webserver kein HTTP/3 spricht?
  - Ja, denn der HTTPS-Record wird von modernen Browsern als erstes abgefragt, erst dann AAAA- und A-Records
  - Ja, denn ohne HTTPS-Record wird ein moderner Browser erst HTTP/3 auf UDP/443 versuchen, und dann auf *legacy* HTTP/HTTPS zurückfallen

# Weitere Informationen zu HTTPS-Records:

- A First Look at SVCB and HTTPS DNS Resource Records in the Wild https://www.net.in.tum.de/fileadmin/bibtex/publications/papers/zir
- Use of HTTPS resource records https://blog.apnic.net/2023/12/18/use-of-https-resource-records/
- Deciphering the Digital Veil: Exploring the Ecosystem of DNS HTTPS Resource Records https://arxiv.org/pdf/2403.15672v1

# I-D: DNS IPv6 Transport Operational Guidelines

# DNS IPv6 Transport Operational Guidelines

- Operational Guidelines for DNS Transport in Mixed IPv4/IPv6 Environments (Best Current Practice, Update zu RFC 3901)
- IPv6 entwickelt sich langsam zum dominanten IP-Protokoll im Internet
- IPv4 wird (noch) benötigt und kann (leider noch) nicht aus den Netzen entfernt werden

# DNS IPv6 Transport Operational Guidelines

# DNS IPv6 Transport Operational Guidelines

- DNS-Server müssen über IPv6 und IPv4 identische Antworten liefern
- die Erreichbarkeit von DNS-Servern (autoritativ und Resolver) sollte über beide IP-Protokolle getestet werden (Monitoring!)
    - oft ist das Monitoring von DNS-Servern immer noch IPv4-only

- IPv6-only autoritative DNS-Server nur dann, wenn sichergestellt ist, das alle potentiellen Nutzer der Domain auch DNS über IPv6 abfragen können (bekannter, geschlossener Benutzerkreis)

# RFC 9715 - Fragmentation Avoidance in DNS over UDP

# Fragmentation Avoidance in DNS over UDP

- DNS-Antworten (ohne DNSSEC-Validierung) können von Angreifern gefälscht werden, wenn die DNS-Antworten über UDP gesendet und aufgrund der Paketgrösse auf dem Transportweg fragmentiert werden
    - siehe BSI: IP Fragmentierung und Maßnahmen gegen "Cache Poisioning"
    - siehe RFC 8900 - IP Fragmentation Considered Fragile (BCP, September 2020)

# Fragmentation Avoidance in DNS over UDP

- RFC 9715 IP Fragmentation Avoidance in DNS over UDP (Informational, Januar 2025) beschreibt verschiedene Einstellungen und Techniken, um Fragmentierung von DNS Nachrichten über UDP zu vermeiden
- Die Umsetzung dieser Empfehlungen ist von Betriebssystem zu Betriebssystem unterschiedlich

# Fragmentation Avoidance in DNS over UDP

- Bei Open-Source DNS-Software sind die meisten dieser Empfehlungen seit Oktober 2020 der "Default"
    - es gibt immer wieder Konfigurationen, bei denen vor vielen Jahren die "Defaults" manuell überschrieben wurden (zum Teil von Linux-Distributionen), und diese Konfigurationen nie an die neuen Entwicklungen angepasst wurden

# Empfehlungen für antwortende DNS-Software

- bei IPv6 auf UDP-Fragmentierung verzichten
- die maximale Paketgrösse sollte aufgrund der vom Anfrager signalisierten EDNS-UDP-Paketgrösse, Netzwerkschnittstellen-MTU, Netzwerk-MTU und der empfolenden maximalen DNS-Paketgrösse 1400 Bytes errechnet werden (kleinster Wert = maximale Grösse von DNS-Nachrichten-Paketen)
- Ist eine Nachricht zu groß für UDP, so soll der Sender dem Anfrager per TC-Bit im Header mitteilen, das die Antwort nochmals über TCP angefragt werden soll

# Empfehlungen für anfragende DNS-Software

# Empfehlungen für Betreiber von DNS-Zonen

- die Grösse von Resource-Record-Sets (A, AAAA, TXT, NS, DNSKEY) sollte 1232 Byte nicht überschreiten
  - gut (155 Bytes):

```
$ dig txt guug.de | grep SIZE
;; MSG SIZE  rcvd: 155
```

  - nicht gut (3562 Bytes):

```
$ dig txt oracle.com | grep SIZE
;; MSG SIZE  rcvd: 3562
```

# Empfehlungen für Betreiber von DNS-Zonen

- im DNS-Server eine *minimal responses* Konfiguration anschalten
  - bei einer *minimal responses* Konfiguration antwortet ein DNS-Server nur mit den angefragten Daten, und gibt keine zusätzlichen Informationen in der Authority-Sektion oder der Additional-Sektion.

- Beispiel BIND 9 `named.conf`:

```
options {
  minimal-answers yes;
  [...]
};
```

# Empfehlungen für Betreiber von DNS-Zonen

- bei DNSSEC gesicherten Zonen: DNSSEC Signatur-Algorithmen verwenden, welche kleine Schlüssel und Signaturen erzeugen
    - ECDSA256/ECDSA384/ED25519/ED448 statt RSA

# RFC 9476 - Die .alt Top-Level-Domain

# Die .alt Top-Level-Domain

- RFC 9476 (Standards Track - September 2023) reserviert die Top-Level-Domain `alt.` für Namensdienste außerhalb des DNS
- In der Vergangenheit wurden von der IETF DNS-Namen für nicht-DNS-Dienste reserviert, wie z.B. `onion.` für das TOR-Netzwerk
- Um Wildwuchs zu vermeiden, kann der `alt`-Namensraum nun ohne Interaktion mit der IETF genutzt werden
- Der `alt`-Namensraum ist unreguliert, jeder kann Namen unterhalb von `alt` für eigene Protokolle verwenden

# .alt TLD und DNS-Server

- DNS-Server (autoritative und DNS-Resolver) sollten Anfragen nach Domain-Namen unterhalb von `alt` ignorieren und **nicht** auflösen
- Betreiber von Domain-Servern sollten keine DNS-Zonen unterhalb von `.alt` anlegen
- Betreiber von Resolvern sollten Anfragen nach DNS-Namen unterhalb von `.alt` stoppen (lokale leere Zone oder durch eine RPZ-Zone)

# I-D: A Top-level Domain for Private Use

# I-D: A Top-level Domain for Private Use

- Seit RFC 1918 (BCP, Februar 1996, aktualisiert durch RFC 6761) sind IPv4-Adress-Bereiche für die Benutzung in internen Netzen reserviert (192.168/16, 10/8, ...)
    - Für DNS gibt es bisher keinen reservierten Namensraum für lokale Netze
    - `.local` ist für Multicast-DNS (mDNS) reserviert
    - `home.arpa` ist für die Benutzung in Heimnetzen mit dem HomeNet Protokoll vorgesehen

# I-D: A Top-level Domain for Private Use

- In vielen Firmen und Organisationen gibt es einen Wildwuchs von "lokalen" DNS-Namen, zum Teil mit Sicherheits- und operativen Problemen
- Die Top-Level `internal` wurde von der ICANN als TLD für den Gebrauch in lokalen/privaten Netzen reserviert
- Ein Internet Draft draft-davies-internal-tld beschreibt die Benutzung dieser reservierten `internal` TLD
- Auch wenn der I-D noch in Bearbeitung ist, so ist die Domain `internal` schon reserviert und kann heute für interne Netze benutzt werden

# RFC 8914 - Extended DNS errors (EDE)

# Traditionelle DNS-Fehlermeldungen

- Für die Übermittlung von Fehlerzuständen sah das original DNS-Protokoll (RFC 1034/1035) ein 4-Bit Feld (*RCODE*) im DNS-Header vor
    - 4-Bit = 16 Mögliche Fehlerzustände

# DNS Paket-Header

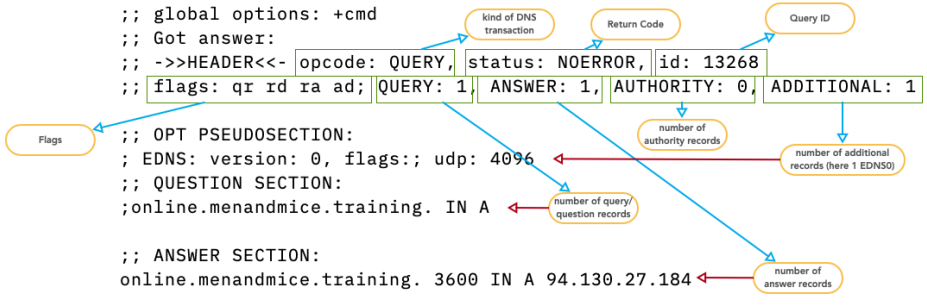
```
% dig online.menandmice.training +multi

; <<>> DiG 9.16.2 <<>> online.menandmice.training +multi
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13268
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;online.menandmice.training. IN A

;; ANSWER SECTION:
online.menandmice.training. 3600 IN A 94.130.27.184

;; Query time: 69 msec
;; SERVER: 172.22.1.8#53(172.22.1.8)
;; WHEN: Sun May 24 11:10:00 CEST 2020
;; MSG SIZE  rcvd: 71
```



# DNS-Fehlerzustände

- **status/rcode**:
  - `NOERROR` - Abfrage war erfolgreich
  - `NXDOMAIN` - der angefragte Domain-Name existiert nicht (oder die Domain ist nicht delegiert)
  - `SERVFAIL` - ein Fehler auf dem Server ist aufgetreten
  - `FORMERR` - die Anfrage war kein korrektes DNS-Paket
  - `REFUSED` - dieser Server darf diese Anfrage nicht beantworten
  - `NOTIMPL` - diese Funktion ist nicht implementiert
  - `BADCOOKIE` - Server Cookie fehlt/nicht korrekt

... siehe IANA Registry für DNS RCODEs

# EDNS

- Die EDNS-Erweiterung (Extended DNS) OPT-Record (RFC 6891 Extension Mechanisms for DNS: EDNS(0), Standards Track, August 1999/April 2013) erweitert den DNS-Header um weitere Felder
- DNS-Software mit EDNS-Unterstützung sendet Meta-Informationen in einem OPT-Resource-Record innerhalb der Additional-Sektion einer DNS-Nachricht
- Der OPT-Record wird dynamisch bei Sender erzeugt und beim Empfänger entfernt
- Der OPT-Record taucht nicht in Zonen-Dateien auf

# Funktionen von EDNS

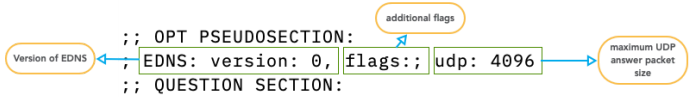
```
% dig online.menandmice.training +multi

; <<>> DiG 9.16.2 <<>> online.menandmice.training +multi
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13268
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;online.menandmice.training. IN A

;; ANSWER SECTION:
online.menandmice.training. 3600 IN A 94.130.27.184

;; Query time: 69 msec
;; SERVER: 172.22.1.8#53(172.22.1.8)
;; WHEN: Sun May 24 11:10:00 CEST 2020
;; MSG SIZE  rcvd: 71
```



# Funktionen von EDNS

- EDNS Version 0 ist die aktuelle Version
    - neue Versionen werden/wurden in der IETF diskutiert, aber noch nicht standardisiert

- Zusätzliche EDNS flags. **DO** - DNSSEC OK, dieser DNS client versteht DNSSEC-Records
- Maximale Grösse für DNS Antworten über UDP, möglich 512 bis 4096 Bytes
- Default seit dem dem *DNS flag day 2020* (Oktober 2020) ist 1232 Byte

# Extended DNS errors (EDE)

- RFC 8914 - Extended DNS Errors ("EDE", Standards Track, Oktober 2020) definiert eine Methode zur Übermittlung zusätzlicher Fehlerinformationen in einer DNS-Antwort.
- Diese neue Funktion ist in `dig` seit Version 9.16.4 und in BIND 9 seit Version 9.18 implementiert
- Auch andere Open-Source-DNS-Server und -Resolver, sowie DNS-Resolver im Internet (Cloudflare) liefern EDE-Fehlermeldungen
- Blog-Post: Extended DNS Errors used in DNS software and services

# Extended DNS errors (EDE)

```
$ dig lame.defaultroutes.org soa @1.1.1.1

; <<>> DiG 9.18.41 <<>> lame.defaultroutes.org soa @1.1.1.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 29410
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDIT

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; EDE: 22 (No Reachable Authority): (at delegation lame.defa
;; QUESTION SECTION:
;lame.defaultroutes.org.                IN      SOA

;; Query time: 705 msec
;; SERVER: 1.1.1.1#53(1.1.1.1) (UDP)
;; WHEN: Mon Nov 03 13:19:33 CET 2025
;; MSG SIZE  rcvd: 94
```

# RFC 9567 - DNS/DNSSEC-Fehler-Meldungskanal

# DNS/DNSSEC DNS-Fehler-Meldungskanal

- Der RFC 9567 - DNS Error Reporting (Standards Track, April 2024) beschreibt eine Möglichkeit für DNS-Resolver-Software, Fehlerzustände bei der DNS-Namensauflösung an die Betreiber der autoritativen DNS-Zone zurückzumelden
- Diese Funktion ähnelt der DMARC-Berichterstattung in E-Mails
- Die gemeldeten Fehler basieren auf "Extended DNS Errors (EDE, RFC 8914)"

# DNS/DNSSEC-Fehler-Meldungskanal

- Die Fehlermeldungen werden innerhalb des DNS-Protokolls versendet
    - Autoritative DNS-Server können Fehler-Reports mittels Opt-In innerhalb einer EDNS-Option anfordern
    - Die autoritativen DNS-Server können wählen, ob nur Fehler-Situationen oder ob auch positives Feedabck (erfolgreiche Namensauflösung) gemeldet werden soll

# DNS/DNSSEC-Fehler-Meldungskanal

- Um einen Fehler zu melden, kodiert der DNS-Resolver den Fehlerreport im **QNAME** (Domain-Name) der Report-DNS-Anfrage.
- Der Resolver baut den **QNAME** aus dem DNS-Label `_er`, dem EDE-Fehlercode, dem **QTYPE** (DNS-Record-Type der fehlerhaften Anfrage), dem **QNAME** der Anfrage, dem DNS-Label `_er` (nochmals) und dem Reporting-Domain-Namen.

# DNS/DNSSEC-Fehler-Meldungskanal

- Der resultierende Domain-Name (**QNAME**) wird dann benutzt um eine DNS-Anfrage für den DNS-Record-Typ **NULL** anzufragen. Über den Domain-Namen wird der Fehler-Report gesendet.

# DNS/DNSSEC-Fehler-Meldungskanal

- Die Domain für Fehlermeldungen ist in der Regel unterschiedlich von der Produktions-Domain.
    - Die Domain für Fehlerreport sollte **nicht** DNSSEC signiert sein, denn nur die Anfragen sind wichtig, die Antworten aus dieser Domain werden ignoriert und müssen nicht abgesichert werden
    - Der autoritative DNS-Server wird mit `NODATA` / `NXRRSET`. Diese negative Antwort wird gecached
        - Die TTL für negative Antworten (der Fehler-Report-Domain) bestimmt, in welcher Frequenz Fehlerreport für den gleichen Fehler gesendet werden, da der DNS-Resolver einen neuen Report erst sendet sobald die TTL abgelaufen ist

# DNS/DNSSEC-Fehler-Meldungskanal

# DNS/DNSSEC-Fehler-Meldungskanal

- Der DNS-Resolver baut einen Domain-Namen *QNAME* für den Fehler-Report `_er.7.1.broken.test._er.a01.reporting-agent.example` und sendet die DNS-Anfrage. Dieser *QNAME* signalisiert das ein DNS Fehler 7 (abgelaufende DNSSEC-Signatur) bei der DNS-Auflösung eines A-Records (*QTYPE 1*) für die Domain `broken.test` aufgetreten ist

# DNS/DNSSEC-Fehler-Meldungskanal

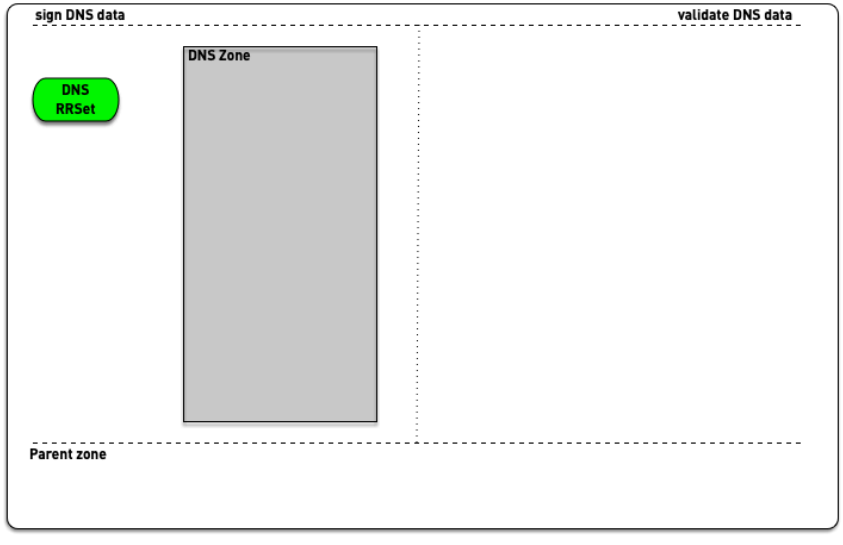
- Die Anfrage wird von einem der autoritativen DNS-Server der Fehler-Report-Domain empfangen und interpretiert.
- Der Fehler-Report-Agent informiert (per Logfile, E-Mail etc) den Betreiber der Domain `broken.test`, das bei einem DNS-Resolver ein DNS-Fehler 7 bei der Auflösung eines A-Record in dieser Domain aufgetreten ist.
- Der Betreiber freut sich über den Report (oder nicht) und behebt den Fehler (oder nicht)

# DNS/DNSSEC-Fehler-Meldungskanal

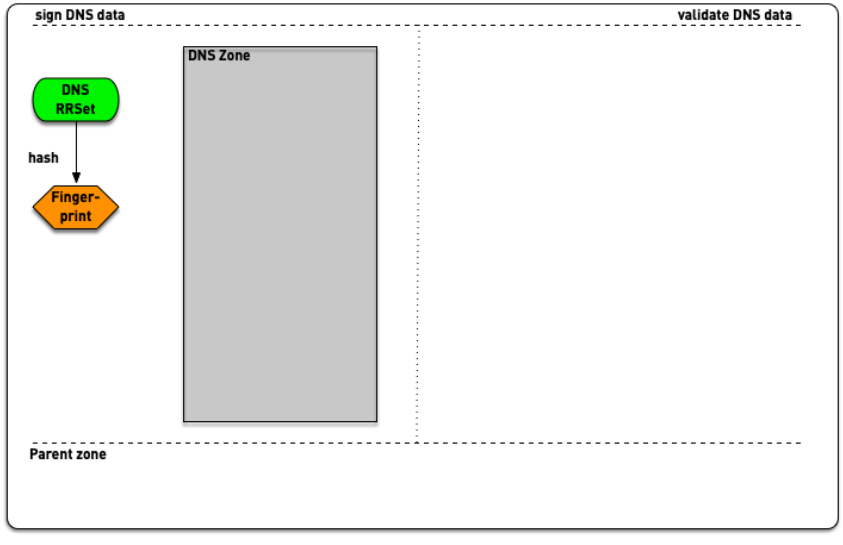
- Der DNS-Fehlerkanal ist derzeit in Unbound (Version 1.23 - April 2025) verfügbar und wird im kommenden BIND 9.22.x (Q2 2026) Einzug halten.

# I-D: "Dry-Run" DNSSEC

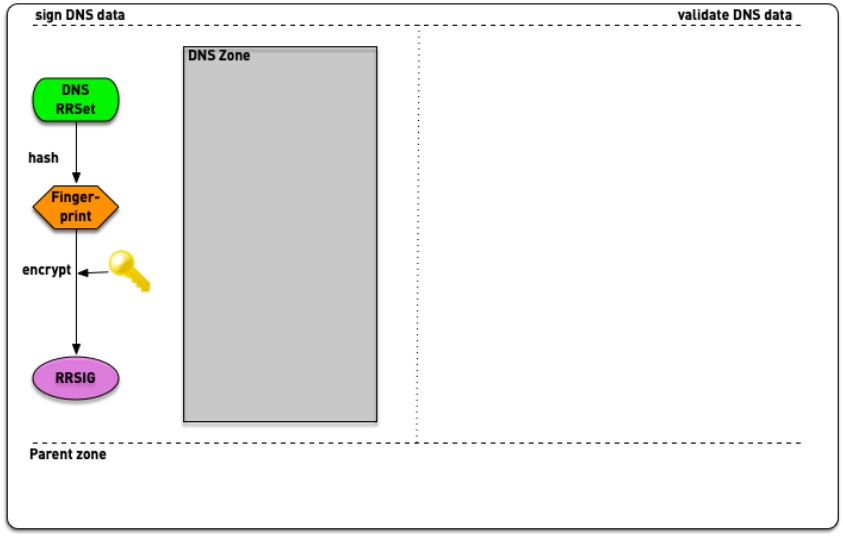
# DNSSEC in a nutshell



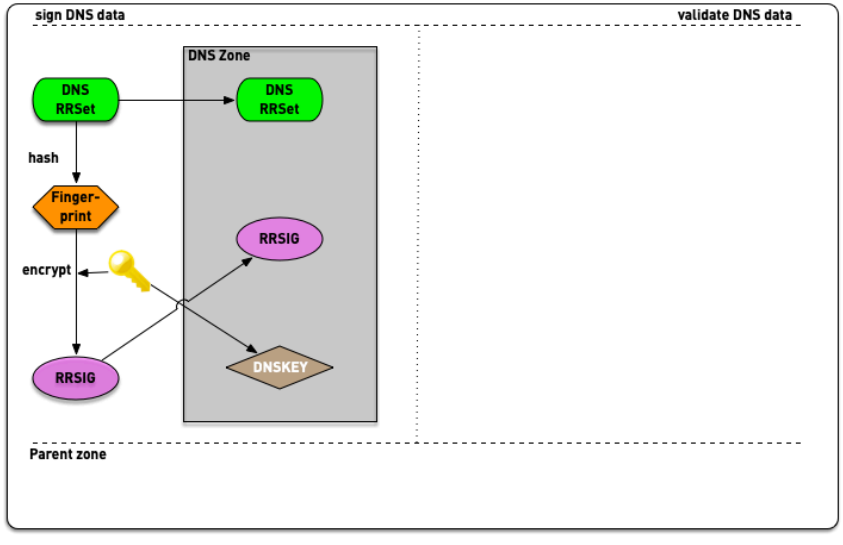
# DNSSEC in a nutshell



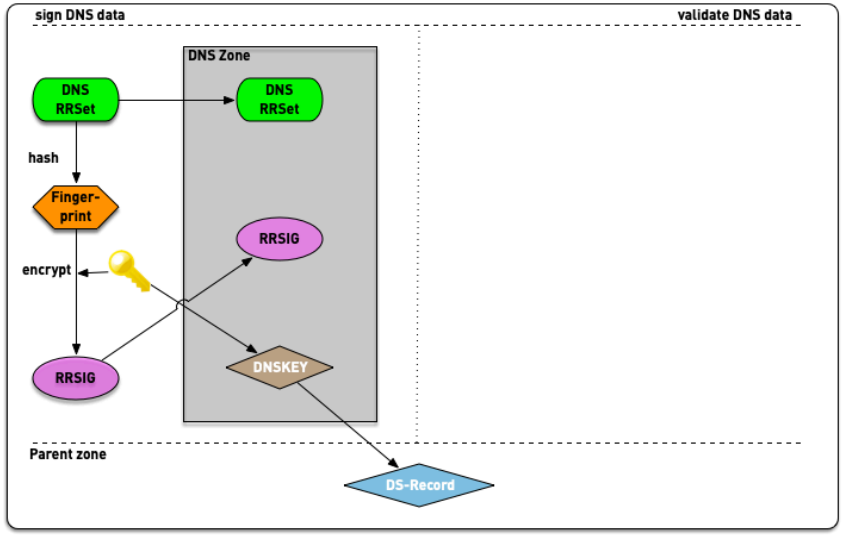
# DNSSEC in a nutshell



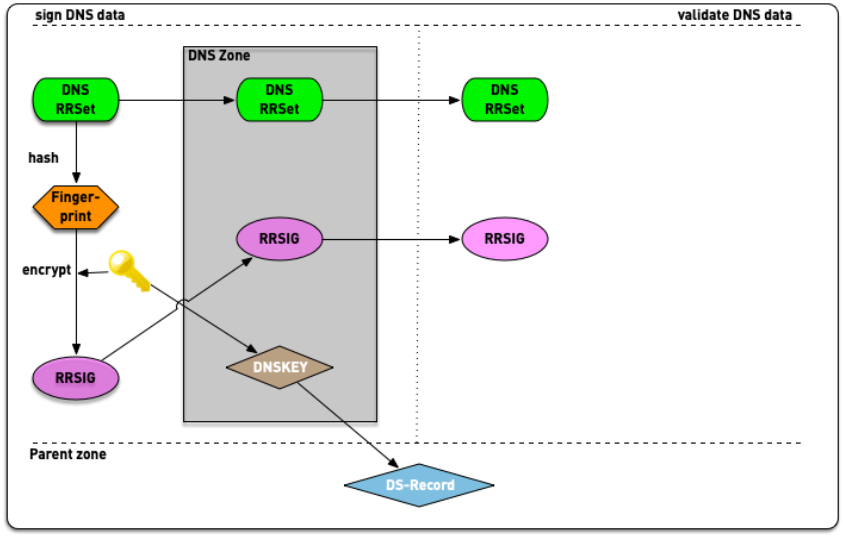
# DNSSEC in a nutshell



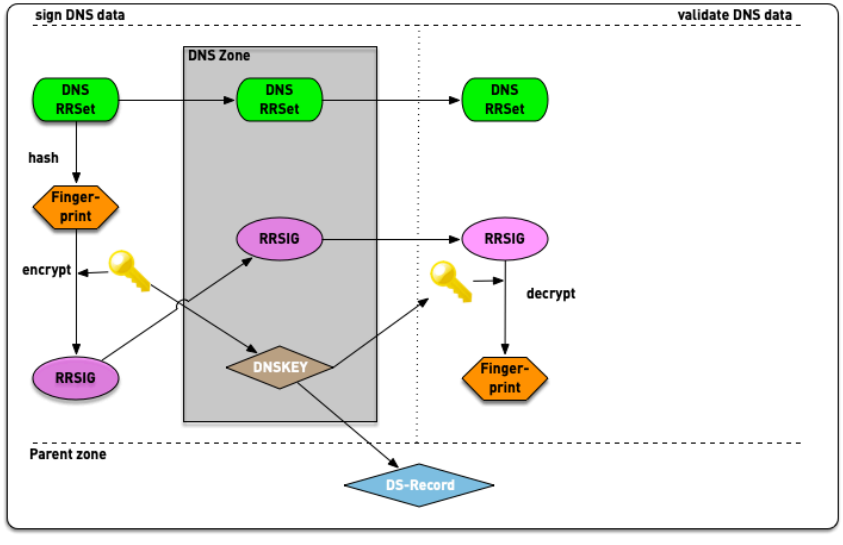
# DNSSEC in a nutshell



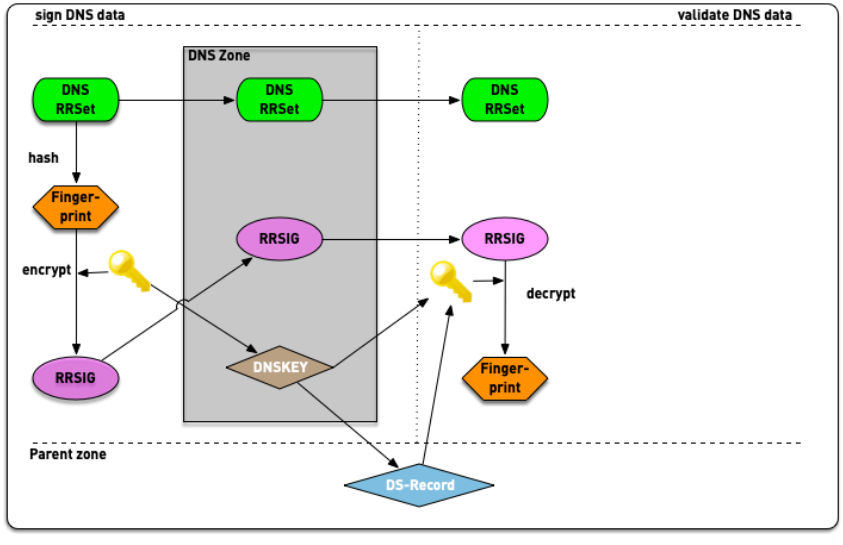
# DNSSEC in a nutshell



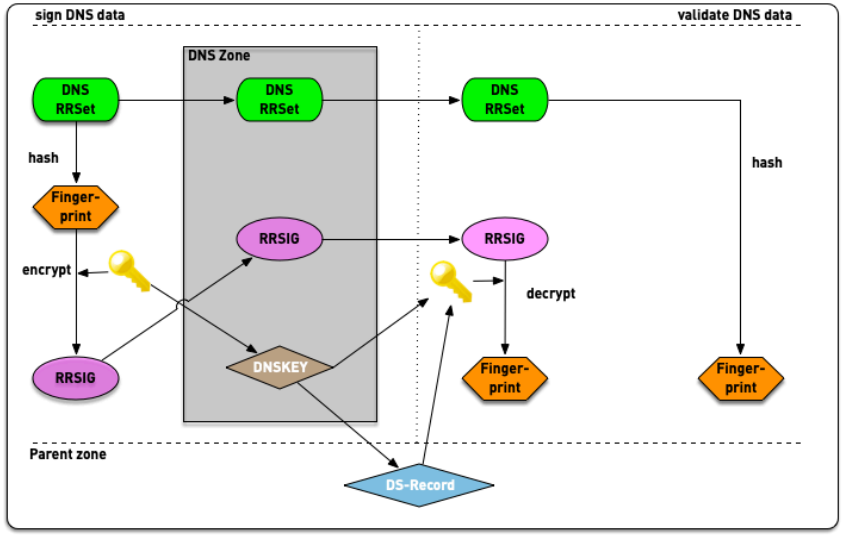
# DNSSEC in a nutshell



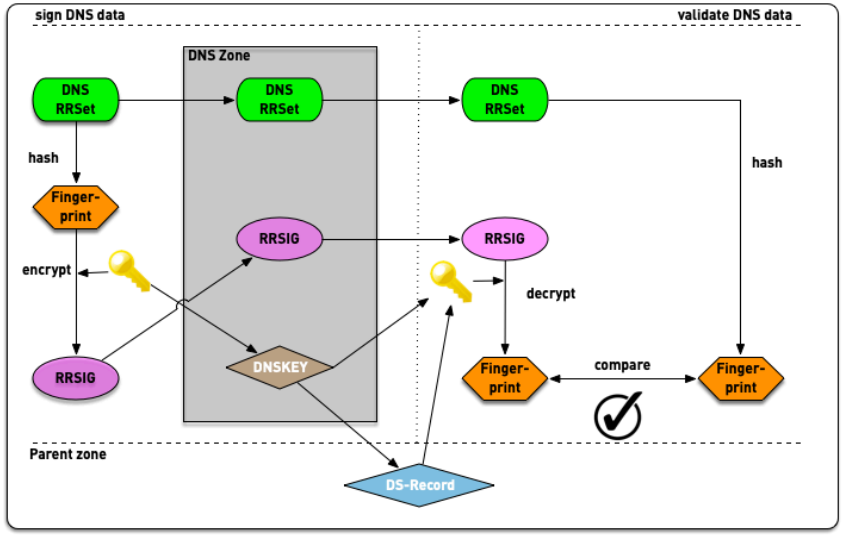
# DNSSEC in a nutshell



# DNSSEC in a nutshell



# DNSSEC in a nutshell



# Problem mit DNSSEC

- DNSSEC wird über den DS-Record in der Eltern-Zone (meist eine TLD) an- oder ausgeschaltet
    - DS-Record in Elternzone = DNSSEC aktiv
    - DS-Record nicht in Elternzone = DNSSEC nicht aktiv

Es gibt keinen Mechanismus, um eine DNSSEC signierte Zone vor der Aktivierung von DNSSEC umfassend zu testen

# "Dry-Run" DNSSEC

- Der Internet-Draft Dry-Run-DNSSEC definiert eine Methode, um DNSSEC vor der Aktivierung umfassend zu testen
    - Dieses Dokument ist kein RFC-Standard, könnte aber einer werden
    - "Dry-Run-DNSSEC" basiert auf den RFCs ***Extended DNS Errors*** (RFC8914) und ***DNS Error Reporting*** (RFC9567).

# "Dry-Run" DNSSEC - die Idee

- Ein DS-Record mit einem bisher nicht benutzten Hashing-Mode als Signal für "Dry-Run-DNSSEC"
  - DNS-Resolver versuchen die DNS-Daten zu validieren
  - Bei erfolgreicher Validierung werden die Daten mit **AD**-Flag (Authentic Data) an den DNS-Client geliefert
  - Bei einem Fehler in der Validierung wird die Antwort ohne **AD**-Flag an den Client gesendet, und ein Fehlerreport an den Reporting-Agent der Zone

# "Dry-Run" DNSSEC - die Idee

- Der Besitzer oder Betreiber der Domain kann DNSSEC für einige Zeit testen und Fehler-Reports sammeln
- Bei breiter Unterstützung dieses Protokolls in DNS-Resolvern und bei wenig oder keinen gemeldeten Fehlern kann DNSSEC auf der Zone aktiviert werden

# "Wet-Run DNSSEC" Opt-In

- Über eine neue (noch zu definierende) EDNS-Option kann ein DNS-Client die DNSSEC-Validierung auf einer Domain mit "Dry-Run-DNSSEC" DS-Record selektiv freischalten
- Hierdurch ist ein Test der DNSSEC-Validierung via Opt-In pro Anfrage möglich

# YAML Output in `dig`

# YAML Output in `dig`

- Neue Versionen des Programms `dig` (Version 9.16+) können die DNS Daten im *YAML* Format ausgeben

```
$ dig a example.com +yaml
```

- *YAML* ist einfacher in Skripten oder Programmen zu parsen (verglichen mit der normalen `dig` Ausgabe, und ggf. auch einfacher für Menschen zu lesen)

- Mit dem Programm `yq` kann zwischen *YAML* und *JSON* umgewandelt werden

# YAML Output in `dig` (Beispiel)

```
$ dig a example.com +yaml
- type: MESSAGE
  message:
    type: AUTH_RESPONSE
    query_time: !!timestamp 2025-10-30T15:41:26.326Z
    response_time: !!timestamp 2025-10-30T15:41:26.339Z
    message_size: 147b
    socket_family: INET
    socket_protocol: UDP
    response_address: "192.0.2.53"
    response_port: 53
    query_address: "0.0.0.0"
    query_port: 0
    response_message_data:
      opcode: QUERY
      status: NOERROR
      id: 36989
      flags: qr aa rd
      QUESTION: 1
      ANSWER: 1
      AUTHORITY: 2
      ADDITIONAL: 1
      OPT_PSEUDOSECTION:
        EDNS:
          version: 0
          flags:
          udp: 1220
          COOKIE:
            CLIENT: 43c62d22d8149b80
            SERVER: 01000000690387265fec604c141dc62d
            STATUS: good
          EDE:
            INFO-CODE: 18 (Prohibited)

      QUESTION_SECTION:
        - 'example.com. IN A'
      ANSWER_SECTION:
        - 'example.com. 3600 IN A 192.0.2.10'
      AUTHORITY_SECTION:
        - 'example.com. 1800 IN NS ns3.myinfrastructure.org.'
        - 'example.com. 1800 IN NS ns5.myinfrastructure.org.'
```

# RFC 9606 - DNS Resolver Information

# RFC 9606 - DNS Resolver Information

- RFC 9606 definiert einen neuen DNS Query-Typ **`RESINFO`** (DNS Resolver Info)
- Wird der DNS-Name `resolver.arpa` abgefragt, so antwortet der DNS-Resolver mit Informationen über die Fähigkeiten des Resolvers:
  - `qnamemin` - Unterstützung für QNAME-Minimization
  - `exterr` - unterstützte EDE-Fehlercodes
  - `infourl` - HTTPS-URL mit Informationen zur Fehlersuche. Gedacht für den IT-Support, nicht für den Endanwender

# RFC 9606 - DNS Resolver Information

- DNS Clients können die **`RESINFO`** mehrerer verfügbarer DNS-Resolver vergleichen und eine Auswahl treffen
    - basierend auf Privatsphäre-Funktionen, z.B. QNAME-Minimization
    - basierend auf Blocklisten auf dem DNS-Resolver

# RFC 9606 - DNS Resolver Information (Beispiel)

```
# dig @192.0.2.101 resolver.arpa RESINFO

; <<>> DiG 9.20.13 <<>> resolver.arpa resinfo @192.0.2.101
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56228
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 185d3f0c07fa57330100000069038b052bd2f6ab81dca9b0 (good)
;; QUESTION SECTION:
;resolver.arpa.                 IN      RESINFO

;; AUTHORITY SECTION:
resolver.arpa.    7200  IN  RESINFO  "qnamemin exterr=15-17 infourl=https://resolver.example.com/guide"

;; Query time: 7 msec
;; SERVER: 192.0.2.101#53(192.0.2.101) (UDP)
;; WHEN: Thu Oct 30 16:57:57 CET 2025
;; MSG SIZE  rcvd: 118
```

# RFC 9859 - Generalized DNS-Notifications

# Notify

- DNS erlaubt es autoritative DNS-Servern, bei einem Update einer DNS-Zone (SOA-Serial-Nummer wurde erhöht), ein Notify-Paket an die anderen autoritatven DNS-Server der DNS-Zone zu senden
    - Die anderen DNS-Server der Zone werden so über die neue Version der Zone informiert und können zeitnah per Zonentransfer die neue Version der Zone laden
    - DNS-Notify Meldungen sind DNS-Nachrichten über UDP/53 (Notify kann auch über TCP/53 gesendet werden, in der Regel wird UDP verwendet).

# Notify

- Das original DNS-Notify wurde in RFC1996 (08/1996) spezifiziert
- Dieses DNS-Notify kann nur zur Information über neue Zonenversionen benutzt werden

# Neue Einsatzgebiete für DNS-Notify

- Der Austausch eines DNSSEC-Delegation Signer bei einem DNSSEC-Key-Rollover kann über das in "Managing DS Records from the Parent via CDS/CDNSKEY" (RFC 8078, Standards Tack, März 2017) definiert Protokoll automatisiert werden
    - Es gibt derzeit jedoch keine Möglichkeit für einen autoritativen DNS-Server, einen DNSSEC-Key-Rollover der Eltern-Zone zu melden
    - Bisherige Lösung: die DNS-Server der Eltern-Server fragen **alle** DNSSEC-signierten Kind-Zonen periodisch ab – Skalierungsproblem!

# Generalized DNS-Notifications

- RFC 9856 definiert einen generischen Notify-Mechanismus für DNS
- Generische Notify-Pakete müssen nicht an DNS-Server der Eltern-Zone gesendet werden:
    - Ein DSYNC-Record in der Zone informiert über DNS-Notify Empfänger für die Zone

# Syntax eines Wildcard DSYNC-Record

- Ein Wildcard-DSYNC-Record informiert über die Server für alle Kind-Zonen dieser Zone:

```
*._dsync.example.  IN DSYNC  CDS   NOTIFY port target
*._dsync.example.  IN DSYNC  CSYNC NOTIFY port target
```

# Beispiel eines spezifischen DSYNC-Record

- Ein spezifischer DSYNC-Record informiert über die Server für eine spezifische Kind-Zone (hier `guug.de`) dieser Zone (`de` TLD):

```
guug._dsync.de.  IN DSYNC  CDS   NOTIFY 53 notify.denic.de.
guug._dsync.de.  IN DSYNC  CSYNC NOTIFY 53 notify.denic.de.
```

# Hausaufgabe: DNS Audit

- Qualität der eigenen DNS-Systeme prüfen und verbessern
  - https://internet.nl
  - https://dnsaudit.io
  - https://zonemaster.net

# Hausaufgabe: RFC 8567 - Customer Management over DNS

# RFC 8567 - Customer Management over DNS

- RFC 8667 (Informatial, April 2019) definiert ein paar bemerkenswerte neue DNS-Resource Records
    - `PASSWORD` Passwort eines CPE-Gerätes
    - `CREDITCARD` Kreditkarten-Daten eines Kunden
    - `SSN` Social Security Nummer der Kunden
    - `SSNPTR` Rückwärtsauflösung einer *Social Security Nummer* zum Namen des Kunden

# FIN

- Fragen? Antworten!