3.8 DNS-Fehler systematisch analysieren

Das Domain Name System ordnet Namen technischen Informationen zu. Dazu gehören insbesondere IPv4- und IPv6-Adressen, zuständige Nameserver, Mailserver, Dienstinformationen und Validierungsdaten.

Ein DNS-Fehler liegt nahe, wenn ein Dienst über seine IP-Adresse erreichbar ist, aber nicht über seinen Namen. Dieser Vergleich ist jedoch nur dann aussagekräftig, wenn der Dienst einen direkten Zugriff über die IP-Adresse technisch unterstützt. HTTPS, virtuelle Webserver und Load Balancer benötigen häufig zusätzlich den korrekten Hostnamen.

> **Grundregel:** Zuerst ermitteln, welcher Name von welchem Client über welchen Resolver aufgelöst werden soll. Danach Antwort, Fehlerstatus, Autorität, Cache, Transportweg und tatsächliche Nutzung durch die Anwendung prüfen.

---

1. Welche Aufgaben erfüllt DNS?

DNS kann unter anderem folgende Informationen bereitstellen:

Recordtyp	Aufgabe
A	Name zu IPv4-Adresse
AAAA	Name zu IPv6-Adresse
CNAME	Alias auf einen anderen Namen
MX	zuständige Mailserver
NS	autoritative Nameserver einer Zone
SOA	grundlegende Verwaltungsinformationen einer Zone
PTR	IP-Adresse zu Name bei Reverse DNS
TXT	frei definierte Textinformationen und Verifikationsdaten
SRV	Dienst, Zielhost, Port, Priorität und Gewichtung
CAA	zulässige Zertifizierungsstellen
DS	Verweis in der DNSSEC-Vertrauenskette
DNSKEY	öffentlicher DNSSEC-Schlüssel
RRSIG	DNSSEC-Signatur eines Recordsets

Beispiel

```
app.example.intern
  → CNAME proxy.example.intern
    → A 192.0.2.20
    → AAAA 2001:db8:20::20
```

Die Anwendung kann am Ende IPv4 oder IPv6 verwenden. Daher müssen alle beteiligten Records und der tatsächlich gewählte Zielendpunkt berücksichtigt werden.

2. Welche Symptome sprechen für ein DNS-Problem?

- Name kann nicht aufgelöst werden,
- IP-Adresse funktioniert, Name jedoch nicht,
- falsche IP-Adresse wird zurückgegeben,
- interne Namen funktionieren nur im Firmennetz,
- Auflösung funktioniert ohne VPN, aber nicht mit VPN,
- Auflösung funktioniert mit VPN, aber nicht ohne VPN,
- einzelne Clients erhalten andere Antworten,
- Antwort wechselt zwischen alten und neuen Adressen,
- IPv4 funktioniert, IPv6-Verbindung schlägt fehl,
- Reverse-Auflösung fehlt,
- DNS-Abfragen laufen in ein Timeout,
- Antwort lautet `NXDOMAIN`, `SERVFAIL` oder `REFUSED`,
- Suchsuffix erzeugt einen falschen vollständigen Namen,
- Anwendung verwendet trotz erfolgreicher manueller Abfrage eine alte Adresse,
- Browser und Terminal liefern unterschiedliche Ergebnisse,
- nur bestimmte Recordtypen schlagen fehl,
- DNSSEC-validierende Resolver lehnen eine Zone ab,
- ein nicht vorgesehener DNS-Server wird verwendet,
- lokale Hosts-Datei überschreibt die DNS-Antwort.

3. Welche Informationen müssen vor der Analyse erfasst werden?

Information	Beispiel
betroffener Client	`CLIENT-023`
Betriebssystem	Windows 11
betroffener Name	`app.example.intern`
vollständig qualifizierter Name	`app.example.intern.`
erwarteter Recordtyp	`A` und `AAAA`

Information	Beispiel
erwartete IPv4-Adresse	`192.0.2.20`
erwartete IPv6-Adresse	`2001:db8:20::20`
verwendeter DNS-Server	`192.0.2.53`
vorgesehener DNS-Server	`192.0.2.53`
DNS-Suchdomäne	`example.intern`
VPN aktiv	Ja
Fehlerbeginn	`2026-07-31 09:42 CEST`
letzter funktionierender Zeitpunkt	`2026-07-31 09:35 CEST`
exakte Fehlermeldung	Name konnte nicht aufgelöst werden
betroffene Anwendung	Browser
Vergleichsclient	`CLIENT-024` funktioniert

Zusätzlich sollte geprüft werden:

- IPv4- und IPv6-Konnektivität zum Resolver,
- Proxy- und VPN-Konfiguration,
- lokale Hosts-Datei,
- Browserfunktionen wie DNS over HTTPS,
- DNS-Cache,
- zuständige Zone und autoritative Server.

4. Wie läuft eine rekursive DNS-Auflösung vereinfacht ab?

```
Client
  → rekursiver Resolver
    → Root-Nameserver
      → TLD-Nameserver
        → autoritativer Nameserver
          → rekursiver Resolver
            → Client
```

Für interne Zonen kann der Ablauf anders sein:

```
Client
  → interner Resolver
    → interne autoritative Zone
      → Client
```

Oder über einen Conditional Forwarder:

```
Client
  → lokaler Resolver
    → bedingte Weiterleitung für partner.example
      → Resolver des Partners
        → autoritative Antwort
```

Caches können den vollständigen Ablauf verkürzen. Eine vorhandene Cacheantwort muss nicht jedes Mal erneut beim autoritativen Server abgefragt werden.

5. Was ist der Unterschied zwischen rekursivem und autoritativem DNS-Server?

Serverrolle	Aufgabe
rekursiver Resolver	ermittelt Antworten im Auftrag des Clients und speichert sie möglicherweise im Cache
autoritativer Nameserver	liefert verbindliche Daten für eine von ihm verwaltete Zone
Forwarder	leitet Abfragen an einen anderen Resolver weiter
Conditional Forwarder	leitet nur bestimmte DNS-Namensräume an definierte Resolver weiter
Stub Resolver	DNS-Komponente auf dem Endgerät, die Resolver befragt

Ein DNS-Server kann abhängig von Produkt und Konfiguration mehrere Rollen übernehmen.

Wichtige Diagnosefrage

```
Ist die falsche Antwort bereits auf dem autoritativen Server vorhanden
oder entsteht sie erst durch Cache, Weiterleitung oder Clientkonfiguration?
```

6. Welche Ports und Transportprotokolle verwendet DNS?

DNS verwendet grundsätzlich:

Transport	Port	Typischer Einsatz
UDP	53	viele gewöhnliche Abfragen und Antworten
TCP	53	große Antworten, Wiederholungen nach Truncation und Zonentransfers
TCP	853	DNS over TLS
TCP	443	DNS over HTTPS
UDP	443	DNS over HTTP/3 kann über QUIC verwendet werden

Wichtig

Eine Firewallregel, die nur UDP 53 erlaubt, kann DNS teilweise funktionieren lassen und bei bestimmten Antworten dennoch Fehler verursachen.

Mögliche Ursachen für den Wechsel zu TCP:

- Antwort ist für den verwendeten UDP-Transport zu groß,
- Antwort wurde als gekürzt markiert,
- DNSSEC vergrößert die Antwort,
- Anwendung oder Resolver verwendet von Anfang an TCP,
- Zonentransfer wird durchgeführt.

7. Wie wird die DNS-Clientkonfiguration unter Windows geprüft?

DNS-Server je Schnittstelle

```
[RO] Get-DnsClientServerAddress |
    Format-Table InterfaceAlias,
                 InterfaceIndex,
                 AddressFamily,
                 ServerAddresses
```

DNS-Clientkonfiguration

```
[RO] Get-DnsClient |
    Format-Table InterfaceAlias,
                 InterfaceIndex,
                 ConnectionSpecificSuffix,
                 RegisterThisConnectionsAddress,
                 UseSuffixWhenRegistering
```

Vollständige IP-Konfiguration

```
[RO][SENS] ipconfig /all
```

Zu prüfen sind:

- DNS-Server,
- verbindungsspezifisches DNS-Suffix,
- primäres DNS-Suffix,
- DNS-Suffixsuchliste,
- Schnittstellenreihenfolge,
- VPN-Adapter,
- virtuelle Adapter.

DNS-over-HTTPS-Konfiguration

Je nach Windows-Version:

```
[RO] Get-DnsClientDohServerAddress
```

Das Cmdlet kann abhängig von Windows-Version und installiertem PowerShell-Modul nicht verfügbar sein.

8. Wie werden DNS-Abfragen unter Windows mit Resolve-DnsName durchgeführt?

Standardabfrage

```
[TEST] Resolve-DnsName app.example.intern
```

A-Record

```
[TEST] Resolve-DnsName app.example.intern -Type A
```

AAAA-Record

```
[TEST] Resolve-DnsName app.example.intern -Type AAAA
```

CNAME

```
[TEST] Resolve-DnsName app.example.intern -Type CNAME
```

MX

```
[TEST] Resolve-DnsName example.com -Type MX
```

SRV

```
[TEST] Resolve-DnsName `
    _ldap._tcp.example.intern `
    -Type SRV
```

PTR beziehungsweise Reverse DNS

```
[TEST] Resolve-DnsName 192.0.2.20 -Type PTR
```

Bestimmten DNS-Server abfragen

```
[TEST] Resolve-DnsName `
    app.example.intern `
    -Type A `
    -Server 192.0.2.53
```

Nur DNS verwenden und zusätzliche lokale Namensauflösungsmechanismen vermeiden

```
[TEST] Resolve-DnsName `
    app.example.intern `
    -Type A `
    -DnsOnly
```

TCP erzwingen

```
[TEST] Resolve-DnsName `
    app.example.intern `
    -Type A `
    -TcpOnly
```

Mit `-TcpOnly` kann geprüft werden, ob DNS über TCP 53 funktioniert.

9. Wie wird nslookup unter Windows, Linux und macOS verwendet?

Einfache Abfrage

```
[TEST] nslookup app.example.intern
```

Bestimmten Server abfragen

```
[TEST] nslookup app.example.intern 192.0.2.53
```

Bestimmten Recordtyp abfragen

```
[TEST] nslookup -type=AAAA app.example.intern 192.0.2.53
```

```
[TEST] nslookup -type=MX example.com 192.0.2.53
```

Reverse DNS

```
[TEST] nslookup 192.0.2.20 192.0.2.53
```

`nslookup` ist weit verbreitet, bildet aber nicht zwingend denselben Auflösungsweg ab wie eine Anwendung oder der native Systemresolver. Für eine vollständige Analyse sollten zusätzlich die betriebssystemspezifischen Werkzeuge verwendet werden.

10. Wie wird die DNS-Clientkonfiguration unter Linux geprüft?

systemd-resolved

```
[RO] resolvectl status
```

Bestimmte Schnittstelle:

```
[RO] resolvectl status eth0
```

Zu prüfen sind:

- aktueller DNS-Server,
- konfigurierte DNS-Server,
- DNS-Domänen,
- Default-Route für DNS,
- DNSSEC-Einstellung,
- DNS-over-TLS-Einstellung,
- schnittstellenbezogene Resolver.

Resolverdatei

```
[RO] cat /etc/resolv.conf
```

Prüfen, ob es sich um einen symbolischen Link handelt:

```
[RO] ls -l /etc/resolv.conf
```

Beispiel:

```
nameserver 127.0.0.53
search example.intern
options edns0 trust-ad
```

`127.0.0.53` kann auf den lokalen Stub Resolver von systemd-resolved verweisen. Der tatsächlich verwendete Upstream-DNS-Server muss dann mit `resolvectl status` bestimmt werden.

NetworkManager

```
[RO][SENS] nmcli device show |
    grep -E 'GENERAL.DEVICE|IP4.DNS|IP6.DNS|IP4.DOMAIN|IP6.DOMAIN'
```

11. Wie werden DNS-Abfragen unter Linux mit resolvectl durchgeführt?

Systemresolver verwenden

```
[TEST] resolvectl query app.example.intern
```

Bestimmten Recordtyp abfragen

```
[TEST] resolvectl query \
    --type=A \
    app.example.intern
```

```
[TEST] resolvectl query \
    --type=AAAA \
    app.example.intern
```

Bestimmte Schnittstelle verwenden

```
[TEST] resolvectl query \
    --interface=eth0 \
    app.example.intern
```

Resolverstatistik

```
[RO] resolvectl statistics
```

Serverfunktionen zurücksetzen beziehungsweise Cache leeren

Diese Befehle verändern den Resolverzustand und gehören nicht zur ersten Analyse:

```
[PRIV][CHANGE] sudo resolvectl reset-statistics
```

```
[PRIV][CHANGE] sudo resolvectl flush-caches
```

Vor dem Leeren sollte die vorhandene Antwort dokumentiert werden.

12. Wie werden DNS-Abfragen mit dig durchgeführt?

`dig` ist nicht auf jedem System standardmäßig installiert.

A- und AAAA-Abfrage

```
[TEST] dig app.example.intern A
```

```
[TEST] dig app.example.intern AAAA
```

Bestimmten Resolver abfragen

```
[TEST] dig @192.0.2.53 app.example.intern A
```

Kurze Ausgabe

```
[TEST] dig +short @192.0.2.53 app.example.intern A
```

Vollständige Antwort mit Kommentaren

```
[TEST] dig @192.0.2.53 app.example.intern A
```

TCP verwenden

```
[TEST] dig +tcp @192.0.2.53 app.example.intern A
```

DNSSEC-Daten anfordern

```
[TEST] dig +dnssec @192.0.2.53 example.com A
```

Autoritative Nameserver einer Zone

```
[TEST] dig @192.0.2.53 example.intern NS
```

SOA-Record

```
[TEST] dig @192.0.2.53 example.intern SOA
```

Reverse-Auflösung

```
[TEST] dig -x 192.0.2.20 @192.0.2.53
```

Delegationsweg verfolgen

```
[TEST] dig +trace example.com
```

`dig +trace` befragt die DNS-Hierarchie schrittweise selbst. Es entspricht nicht dem normalen rekursiven Weg über den konfigurierten Resolver und kann durch Firewalls oder interne Zonen eingeschränkt sein.

13. Welche Felder einer dig-Ausgabe sind wichtig?

Beispielstruktur:

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12345
;; flags: qr rd ra
;; QUESTION SECTION:
;app.example.intern.       IN A

;; ANSWER SECTION:
app.example.intern. 300    IN A 192.0.2.20

;; Query time: 12 msec
;; SERVER: 192.0.2.53#53
```

```
;; WHEN: ...
;; MSG SIZE  rcvd: ...
```

Feld	Bedeutung
`status`	DNS-Antwortstatus
`id`	Transaktions-ID
`qr`	Nachricht ist eine Antwort
`aa`	autoritative Antwort
`tc`	Antwort wurde gekürzt
`rd`	Rekursion wurde angefordert
`ra`	Rekursion ist verfügbar
`ad`	validierende Stelle kennzeichnet Daten als authentifiziert
`cd`	Client fordert deaktivierte DNSSEC-Prüfung
`ANSWER`	eigentliche Antwortrecords
`AUTHORITY`	zuständige beziehungsweise autoritative Informationen
`ADDITIONAL`	zusätzliche hilfreiche Records
`TTL`	verbleibende Cachelebensdauer
`SERVER`	tatsächlich abgefragter Resolver
`Query time`	gemessene Abfragedauer

Das `ad`-Flag muss im Zusammenhang mit dem befragten validierenden Resolver bewertet werden. Es beweist nicht, dass der lokale Client selbst die gesamte DNSSEC-Kette validiert hat.

14. Wie wird die DNS-Konfiguration unter macOS geprüft?

Aktuelle Resolverkonfiguration

```
[RO][SENS] scutil --dns
```

macOS kann mehrere Resolverkonfigurationen parallel besitzen, beispielsweise für:

- allgemeine Anfragen,
- VPN-Domänen,
- bestimmte Suchdomänen,
- Multicast DNS,

- schnittstellenbezogene Namensräume.

DNS-Server eines Netzwerkdienstes

```
[RO] networksetup -getdnsservers "Wi-Fi"
```

Suchdomänen

```
[RO] networksetup -getsearchdomains "Wi-Fi"
```

IP-Konfiguration

```
[RO] networksetup -getinfo "Wi-Fi"
```

Der tatsächliche Netzwerkdienst muss vorher bestimmt werden:

```
[RO] networksetup -listallnetworkservices
```

Hosts-Auflösung über Systemmechanismen

```
[TEST] dscacheutil -q host -a name app.example.intern
```

Dieser Test bildet den macOS-Systemresolver besser ab als eine direkte Abfrage eines einzelnen DNS-Servers mit `dig`.

15. Warum können dig und eine Anwendung unterschiedliche Ergebnisse liefern?

`dig` sendet eine direkte DNS-Abfrage. Eine Anwendung kann dagegen weitere Mechanismen verwenden:

- Systemresolver,
- lokale Hosts-Datei,
- DNS-Cache des Betriebssystems,
- eigener Anwendungscache,
- Browsercache,
- DNS over HTTPS,
- VPN-spezifischer Resolver,
- Multicast DNS,

- Suchsuffixe,
- Proxy,
- Sicherheitssoftware,
- andere Reihenfolge von IPv4 und IPv6.

Beispiel

```
dig @192.0.2.53 app.example.intern
→ 192.0.2.20

Browser
→ verwendet eigenen DoH-Resolver
→ erhält keine interne Adresse
```

Deshalb sollten mindestens zwei Ebenen geprüft werden:

1. direkte DNS-Serverantwort,
2. tatsächliche Auflösung über das Betriebssystem beziehungsweise die Anwendung.

16. Wie wird die lokale Hosts-Datei geprüft?

Ein Eintrag in der Hosts-Datei kann DNS für den betreffenden Namen übersteuern.

Windows

```
[RO][PRIV][SENS] Get-Content `
    "$env:SystemRoot\System32\drivers\etc\hosts"
```

Nicht auskommentierte Zeilen anzeigen:

```
[RO][PRIV][SENS] Get-Content `
    "$env:SystemRoot\System32\drivers\etc\hosts" |
    Where-Object {
        $_.Trim() -and
        -not $_.Trim().StartsWith('#')
    }
```

Linux

```
[RO][SENS] cat /etc/hosts
```

macOS

```
[RO][SENS] cat /etc/hosts
```

Typische Probleme

- veraltete Testadresse,
- falscher Hostname,
- Eintrag wurde von Software ergänzt,
- IPv4- und IPv6-Einträge widersprechen sich,
- Eintrag existiert nur auf einem Client,
- Eintrag besitzt unsichtbare Formatierungsfehler.

Hosts-Dateien dürfen nicht vorschnell geändert werden. Zuerst müssen Ursprung, Zweck und Konfigurationsmanagement geklärt werden.

17. Wie beeinflusst ein DNS-Suchsuffix die Auflösung?

Ein nicht vollständig qualifizierter Name kann durch eine Suchdomäne ergänzt werden.

Eingabe:

```
app
```

Suchdomäne:

```
example.intern
```

Mögliche Anfrage:

```
app.example.intern
```

Bei mehreren Suchdomänen können mehrere Abfragen entstehen:

```
app.berlin.example.intern
app.example.intern
app
```

Prüfung unter Windows

```
[RO] Get-DnsClient |
    Select-Object InterfaceAlias,
                  ConnectionSpecificSuffix
```

Zusätzlich:

```
[RO][SENS] ipconfig /all
```

Linux

```
[RO] resolvectl status
```

```
[RO] cat /etc/resolv.conf
```

macOS

```
[RO][SENS] scutil --dns
```

Sicherer Vergleich

```
Kurzname:
app

Vollqualifizierter Name:
app.example.intern.
```

Der abschließende Punkt kennzeichnet in DNS-Werkzeugen einen absoluten Namen und verhindert dort normalerweise die Ergänzung eines Suchsuffixes.

18. Wie werden DNS-Antwortstatus richtig interpretiert?

Status	Bedeutung
`NOERROR` mit Antwort	Abfrage war erfolgreich und enthält Antwortdaten
`NOERROR` ohne gesuchten Record	Name kann existieren, aber Recordtyp fehlt
`NXDOMAIN`	abgefragter Name existiert laut Antwort nicht
`SERVFAIL`	Server konnte keine gültige Antwort liefern
`REFUSED`	Server verweigert die Abfrage
`FORMERR`	Abfrageformat wurde als fehlerhaft bewertet
`NOTIMP`	Operation wird nicht unterstützt
Timeout	keine rechtzeitige verwertbare Antwort erhalten

Wichtige Unterscheidung

NXDOMAIN:

Der Name existiert laut DNS-Antwort nicht.

NOERROR ohne A-Record:

Der Name kann existieren, besitzt aber keinen A-Record.

SERVFAIL kann unter anderem entstehen durch:

- DNSSEC-Validierungsfehler,
- nicht erreichbaren autoritativen Server,
- fehlerhafte Delegation,
- Timeout bei Weiterleitung,
- Serverfehler,
- beschädigte Zone,
- fehlende Erreichbarkeit über TCP,
- Schleife zwischen Forwardern.

19. Wie wird zwischen NXDOMAIN und fehlendem Recordtyp unterschieden?

A-Record abfragen

```
[TEST] dig @192.0.2.53 app.example.intern A
```

Mögliche Ergebnisse:

Fall 1 – Name existiert nicht

```
status: NXDOMAIN
```

Fall 2 – Name existiert, besitzt aber keinen A-Record

```
status: NOERROR
ANSWER: 0
```

Im Authority-Bereich kann ein SOA-Record erscheinen.

Fall 3 – Name besitzt nur AAAA

```
[TEST] dig @192.0.2.53 app.example.intern AAAA
```

liefert eine IPv6-Adresse.

Diese Unterscheidung ist wichtig, weil das Anlegen eines A-Records bei einem vollständig falschen Namen eine andere Maßnahme wäre als das Ergänzen eines fehlenden Recordtyps.

20. Wie werden CNAME-Ketten untersucht?

Direkte Abfrage

```
[TEST] dig @192.0.2.53 app.example.intern CNAME
```

Vollständige Auflösung

```
[TEST] dig @192.0.2.53 app.example.intern A
```

Mögliche Kette:

```
app.example.intern
  → CNAME proxy.example.intern
    → CNAME proxy-berlin.example.net
```

```
→ A 192.0.2.20
```

Zu prüfen sind:

- existiert jedes Ziel der Kette?
- besitzt das letzte Ziel einen passenden A- oder AAAA-Record?
- verweist ein CNAME auf sich selbst?
- existiert eine Schleife?
- überschreitet die Kette Grenzen zwischen internen und externen Zonen?
- stimmt die TTL der beteiligten Records?
- passt das TLS-Zertifikat weiterhin zum ursprünglich verwendeten Namen?

Der erfolgreiche Abschluss der DNS-Kette beweist noch nicht, dass die Anwendung oder das Zertifikat korrekt ist.

21. Wie wird Split DNS beziehungsweise Split-Horizon DNS geprüft?

Bei Split DNS liefert derselbe Name abhängig vom verwendeten Resolver oder Standort unterschiedliche Antworten.

Interner Resolver

```
app.example.com
→ 192.0.2.20
```

Öffentlicher Resolver

```
app.example.com
→ 203.0.113.20
```

Dies kann beabsichtigt sein.

Vergleichsabfragen

```
[TEST] dig @192.0.2.53 app.example.com A
```

```
[TEST] dig @ÖFFENTLICHER-RESOLVER app.example.com A
```

Ein öffentlicher Resolver darf nur verwendet werden, wenn dies nach Unternehmensrichtlinie zulässig und aus dem Netz erreichbar ist.

Typische Fehler

- VPN verwendet öffentlichen statt internen Resolver,
- interne Zone fehlt auf einem Standort,
- Conditional Forwarder fehlt,
- Browser umgeht internen DNS per DoH,
- Antwort aus öffentlichem Cache wird verwendet,
- interne und externe Records wurden verwechselt.

22. Wie beeinflussen VPN und schnittstellenbezogenes DNS die Auflösung?

Ein VPN kann:

- eigene DNS-Server setzen,
- nur bestimmte Domänen über interne Resolver leiten,
- alle DNS-Anfragen übernehmen,
- Suchdomänen hinzufügen,
- DNS over HTTPS blockieren oder konfigurieren,
- IPv4 und IPv6 unterschiedlich behandeln.

Prüfung vor und nach VPN-Verbindung

Eigenschaft	vor VPN	nach VPN
DNS-Server	dokumentieren	dokumentieren
Suchdomänen	dokumentieren	dokumentieren
Routen zum DNS-Server	dokumentieren	dokumentieren
A-Antwort	vergleichen	vergleichen
AAAA-Antwort	vergleichen	vergleichen
Systemresolver	vergleichen	vergleichen

Windows

```
[RO] Get-DnsClientServerAddress
```

Linux

```
[RO] resolvectl status
```

macOS

```
[RO][SENS] scutil --dns
```

Ein interner DNS-Server kann korrekt eingetragen sein, aber über die aktuelle Routingtabelle nicht erreichbar sein. Deshalb muss zusätzlich die Route zum Resolver geprüft werden.

23. Wie wird die Erreichbarkeit des DNS-Servers geprüft?

Route zum Resolver

Windows:

```
[RO] Find-NetRoute -RemoteIPAddress 192.0.2.53
```

Linux:

```
[RO] ip route get 192.0.2.53
```

macOS:

```
[RO] route -n get 192.0.2.53
```

TCP-Port 53 testen

Windows:

```
[TEST] Test-NetConnection 192.0.2.53 -Port 53
```

Linux und macOS:

```
[TEST] nc -vz -w 5 192.0.2.53 53
```

Dieser Test prüft nur TCP 53.

Tatsächliche DNS-Abfrage über UDP

```
[TEST] dig @192.0.2.53 app.example.intern A
```

Tatsächliche DNS-Abfrage über TCP

```
[TEST] dig +tcp @192.0.2.53 app.example.intern A
```

Auswertung

UDP-Abfrage	TCP-Abfrage	Mögliche Richtung
erfolgreich	erfolgreich	Transport grundsätzlich verfügbar
erfolgreich	fehlerhaft	TCP 53, Firewall oder Serverlistener prüfen
fehlerhaft	erfolgreich	UDP 53, Fragmentierung oder Filter prüfen
fehlerhaft	fehlerhaft	Route, Firewall, Server oder falsche Adresse

24. Wie werden große DNS-Antworten und das TC-Flag untersucht?

Das DNS-Headerflag `TC` bedeutet, dass eine Antwort gekürzt wurde.

Vereinfachter Ablauf:

```
Client → DNS-Abfrage über UDP
Server → gekürzte Antwort mit TC=1
Client → Wiederholung über TCP
Server → vollständige Antwort über TCP
```

dig-Ausgabe über UDP

```
[TEST] dig @192.0.2.53 example.com DNSKEY
```

Vergleich über TCP

```
[TEST] dig +tcp @192.0.2.53 example.com DNSKEY
```

Wireshark-Display-Filter

```
dns.flags.truncated == 1
```

Mögliche Fehlerursachen:

- TCP 53 wird blockiert,
- Fragmentierung oder MTU-Problem,
- Firewall verwirft große UDP-Antworten,
- EDNS wird fehlerhaft behandelt,
- Resolver oder Middlebox arbeitet nicht standardkonform.

25. Wie werden TTL und DNS-Caching interpretiert?

TTL steht für Time to Live und gibt an, wie lange ein DNS-Record gecacht werden darf.

Beispiel

```
app.example.intern. 300 IN A 192.0.2.20
```

TTL:

```
300 Sekunden = 5 Minuten
```

Nach einer Änderung können Resolver die alte Antwort bis zum Ablauf der noch vorhandenen TTL verwenden.

Wichtig

- verschiedene Resolver können unterschiedliche verbleibende TTLs besitzen,
- Anwendungen können zusätzlich eigene Caches verwenden,
- negative Antworten können ebenfalls gecacht werden,
- Browser können eigene DNS-Zustände verwalten,
- Load Balancer können mehrere Adressen liefern,
- alte Antworten müssen nicht auf allen Clients gleichzeitig verschwinden.

Vergleich zu unterschiedlichen Zeitpunkten

```
[TEST] dig @192.0.2.53 app.example.intern A
```

Zu dokumentieren:

- Antwortadresse,
- TTL,
- Server,
- Zeitpunkt,
- Antwortstatus.

26. Was ist negatives DNS-Caching?

Auch negative Antworten wie `NXDOMAIN` oder das Fehlen eines bestimmten Recordtyps können zeitweise gecacht werden.

Mögliche Situation:

```
09:00 Name existiert noch nicht.
09:01 Client fragt und erhält NXDOMAIN.
09:05 Record wird angelegt.
09:06 Client erhält weiterhin die gecachte negative Antwort.
```

Die Dauer des negativen Cachings hängt unter anderem von den Zonen- und Resolverinformationen ab.

Diagnose

- autoritativen Server direkt abfragen,
- rekursiven Resolver abfragen,
- Systemresolver abfragen,
- Antwortstatus und SOA-Informationen vergleichen,
- Zeitpunkt der Recordänderung dokumentieren.

Ein Cache sollte erst geleert werden, nachdem die ursprüngliche Antwort und deren TTL gesichert wurden.

27. Wie wird der DNS-Cache unter Windows geprüft?

Cache anzeigen

```
[RO] Get-DnsClientCache
```

Nach einem Namen suchen:

```
[RO] Get-DnsClientCache |
    Where-Object Entry -Match 'app\.example\.intern'
```

Klassische Anzeige

```
[RO] ipconfig /displaydns
```

Cache exportieren

```
[RO][FILE][SENS] Get-DnsClientCache |
    Export-Csv 'C:\Temp\dns-cache-vorher.csv' `
    -NoTypeInformation `
    -Encoding UTF8
```

Der Cache kann interne Namen und aufgerufene Ziele enthalten und ist daher als sensibel zu behandeln.

28. Wann darf ein DNS-Cache geleert werden?

Das Leeren des Caches verändert den Diagnosezustand und kann eine fehlerhafte Antwort nur vorübergehend verbergen.

Windows

```
[PRIV][CHANGE] Clear-DnsClientCache
```

Alternative:

```
[PRIV][CHANGE] ipconfig /flushdns
```

Linux mit systemd-resolved

```
[PRIV][CHANGE] sudo resolvectl flush-caches
```

Andere Linux-Resolver besitzen eigene Verfahren. Es existiert kein allgemeiner Befehl für alle Linux-Systeme.

macOS

```
[PRIV][CHANGE] sudo dscacheutil -flushcache
```

Anschließend kann abhängig von macOS-Version zusätzlich ein HUP-Signal an den DNS-Responder erforderlich sein:

```
[PRIV][CHANGE] sudo killall -HUP mDNSResponder
```

Diese macOS-Befehle sollten nur nach Prüfung der eingesetzten Version und lokalen Dokumentation verwendet werden.

Vorher sichern

- aktuelle Antwort,
- TTL,
- verwendeter Resolver,
- Betriebssystemcache,
- Anwendungszustand,
- Zeitpunkt.

29. Wie werden autoritative DNS-Daten geprüft?

Zuständige Nameserver ermitteln

```
[TEST] dig @192.0.2.53 example.intern NS
```

SOA-Record prüfen

```
[TEST] dig @192.0.2.53 example.intern SOA
```

Autoritativen Server direkt abfragen

```
[TEST] dig @AUTORITATIVER-SERVER app.example.intern A
```

In der Antwort sollte bei einer autoritativen Antwort typischerweise das `aa`-Flag erscheinen.

Zu prüfen

- richtige Zone,
- richtiger Record,
- korrekte Adresse,
- TTL,
- Seriennummer der Zone,
- Replikationsstand,
- primärer und sekundärer Nameserver,
- Delegation,
- DNSSEC-Signaturen,
- unterschiedliche Antworten mehrerer autoritativer Server.

Wenn autoritative Server unterschiedliche Zonenversionen liefern, können Clients abhängig vom befragten Server unterschiedliche Ergebnisse erhalten.

30. Wie wird die SOA-Seriennummer verwendet?

Der SOA-Record enthält Verwaltungsinformationen einer DNS-Zone, darunter eine Seriennummer.

```
[TEST] dig @DNS-SERVER example.intern SOA
```

Mehrere autoritative Server vergleichen:

```
[TEST] dig @DNS-SERVER-1 example.intern SOA
```

```
[TEST] dig @DNS-SERVER-2 example.intern SOA
```

Mögliche Auffälligkeit

```
Server 1: Serial 2026073105
Server 2: Serial 2026073002
```

Dies kann auf eine noch nicht erfolgte oder fehlgeschlagene Zonenübertragung beziehungsweise Replikation hinweisen.

Die Bedeutung und das Format der Seriennummer werden durch die DNS-Verwaltung der jeweiligen Umgebung bestimmt. Eine höhere dezimale Darstellung ist häufig, aber nicht jede Organisation verwendet dasselbe Schema.

31. Wie werden Delegationsfehler erkannt?

Eine übergeordnete Zone verweist per NS-Records auf die autoritativen Server einer untergeordneten Zone.

```
example.com
  → Delegation für berlin.example.com
    → ns1.berlin.example.com
    → ns2.berlin.example.com
```

Mögliche Fehler:

- falscher Nameserver eingetragen,
- Nameserver ist nicht erreichbar,
- Glue Record fehlt oder ist falsch,
- Child- und Parent-NS-Records unterscheiden sich,
- Firewall blockiert UDP oder TCP 53,
- DNSSEC-DS-Record passt nicht mehr,
- autoritativer Server kennt die delegierte Zone nicht,
- Schleife in der Delegation.

Öffentliche Delegation verfolgen

```
[TEST] dig +trace berlin.example.com
```

Für interne Zonen muss der interne Delegations- und Weiterleitungsweg anhand der Unternehmensarchitektur geprüft werden.

32. Wie werden DNSSEC-Probleme erkannt?

DNSSEC ermöglicht die kryptografische Prüfung von DNS-Daten. Ein Fehler in der Vertrauenskette kann dazu führen, dass validierende Resolver mit `SERVFAIL` antworten.

Abfrage mit DNSSEC-Daten

```
[TEST] dig +dnssec @192.0.2.53 example.com A
```

Validierung testweise durch den Resolver anfragen, aber Checking Disabled setzen

```
[TEST] dig +cdflag @192.0.2.53 example.com A
```

Wenn eine normale Abfrage `SERVFAIL` liefert, dieselbe Abfrage mit `+cdflag` aber Daten zurückgibt, ist ein DNSSEC-Validierungsproblem eine mögliche Hypothese.

Mögliche Ursachen

- abgelaufene Signatur,
- falscher DS-Record,
- Schlüsselwechsel nicht korrekt abgeschlossen,
- inkonsistente autoritative Server,
- falsche Systemzeit,
- beschädigte Vertrauenskette,
- Antwort wird auf dem Weg verändert oder abgeschnitten.

`+cdflag` deaktiviert nicht dauerhaft die Serverkonfiguration. Es kennzeichnet die einzelne Abfrage entsprechend. Das Ergebnis darf nur als Diagnosehinweis verwendet werden.

33. Wie beeinflusst DNS over HTTPS die Fehleranalyse?

Bei DNS over HTTPS werden DNS-Abfragen in HTTPS-Verbindungen übertragen. Dadurch kann eine Anwendung den betriebssystemseitig konfigurierten DNS-Server umgehen.

Mögliche Auswirkungen:

- interne Zonen werden nicht gefunden,
- Unternehmensfilter greifen nicht,
- Browser liefert andere Antworten als `nslookup`,
- DNS-Verkehr ist im gewöhnlichen Port-53-Mitschnitt nicht sichtbar,
- Split DNS funktioniert in der Anwendung nicht,
- zentrale DNS-Protokolle enthalten die Anfrage nicht.

Prüffragen

- verwendet die Anwendung den Systemresolver?
- ist DoH durch Browser, Betriebssystem oder Richtlinie aktiviert?
- welcher DoH-Endpunkt wird verwendet?
- sollen interne Namen vom Systemresolver aufgelöst werden?

- existieren betriebliche Vorgaben für verschlüsseltes DNS?

DoH sollte nicht pauschal deaktiviert werden. Zuerst müssen Unternehmensrichtlinie, Sicherheitskonzept und tatsächlicher Auflösungsweg geprüft werden.

34. Wie wird DNS-Verkehr mit Wireshark untersucht?

Alle klassischen DNS-Nachrichten

```
dns
```

Nur DNS-Anfragen

```
dns.flags.response == 0
```

Nur DNS-Antworten

```
dns.flags.response == 1
```

Bestimmter Name

```
dns.qry.name == "app.example.intern"
```

Bestimmter Recordtyp

A-Record:

```
dns.qry.type == 1
```

AAAA-Record:

```
dns.qry.type == 28
```

NXDOMAIN

```
dns.flags.rcode == 3
```

SERVFAIL

```
dns.flags.rcode == 2
```

Gekürzte Antworten

```
dns.flags.truncated == 1
```

Lange Antwortzeit

```
dns.time > 1
```

`dns.time` ist verfügbar, wenn Wireshark Anfrage und passende Antwort einander zuordnen konnte.

35. Wie wird DNS-Verkehr sicher mitgeschnitten?

Capture-Filter für klassischen DNS-Verkehr

```
port 53
```

Nur bestimmter DNS-Server

```
host 192.0.2.53 and port 53
```

Zeitlich begrenzte Aufnahme

```
[TEST][PRIV][FILE][SENS] dumpcap \
  -i INTERFACE \
  -f "host 192.0.2.53 and port 53" \
  -a duration:30 \
  -w dns-diagnose.pcapng
```

TShark-Liveanzeige

```
[TEST][PRIV][SENS] tshark \
  -i INTERFACE \
  -f "host 192.0.2.53 and port 53"
```

Felder aus einer Datei ausgeben

```
[RO][SENS] tshark \
  -r dns-diagnose.pcapng \
  -Y "dns" \
  -T fields \
  -e frame.time \
  -e ip.src \
  -e ip.dst \
  -e udp.srcport \
  -e udp.dstport \
  -e dns.id \
  -e dns.flags.response \
  -e dns.flags.rcode \
  -e dns.qry.name \
  -e dns.qry.type
```

DNS-Aufzeichnungen können aufgerufene interne und externe Namen offenlegen und müssen gemäß Seite 2.14 geschützt werden.

36. Wie wird eine DNS-Anfrage im Paketmitschnitt ausgewertet?

Erfolgreicher Ablauf

```
Client → Resolver:
Query A app.example.intern

Resolver → Client:
Response NOERROR
A 192.0.2.20
```

Timeout

Client → Resolver:

Query A app.example.intern

Client → Resolver:

Wiederholung

Keine Antwort

NXDOMAIN

Client → Resolver:

Query A app.example.intern

Resolver → Client:

Response NXDOMAIN

TCP-Fallback

Client → Resolver über UDP:

Query

Resolver → Client:

Response, Truncated

Client → Resolver über TCP:

SYN, SYN-ACK, ACK

DNS Query

Resolver → Client über TCP:

vollständige DNS Response

Zu prüfen

- Transaktions-ID,
- Quell- und Zieladresse,
- UDP oder TCP,
- abgefragter Name,
- Recordtyp,
- Antwortstatus,

- Antwortrecords,
- TTL,
- Wiederholungen,
- Antwortdauer,
- Truncation,
- DNSSEC-Flags.

37. Wie wird zwischen Client-, Resolver- und Zonenproblem unterschieden?

Beobachtung	Wahrscheinlichere Richtung
nur eine Anwendung betroffen	Anwendungscache, DoH oder Proxy
nur ein Client betroffen	Clientcache, Hosts-Datei oder DNS-Konfiguration
alle Clients eines DNS-Servers betroffen	Resolver, Forwarder oder dessen Netzwerkpfad
direkter autoritativer Server korrekt, Resolver falsch	Cache oder Weiterleitung
alle autoritativen Server falsch	Zonendaten
ein autoritativer Server weicht ab	Replikation oder Zonentransfer
UDP funktioniert, TCP nicht	Firewall oder TCP-Listener
TCP funktioniert, UDP nicht	UDP-Filter, Fragmentierung oder Middlebox
interne Namen nur ohne DoH	Anwendung umgeht internen Resolver
Auflösung korrekt, Verbindung fehlerhaft	Problem liegt nach DNS
A funktioniert, AAAA führt zum Fehler	IPv6-Pfad oder Ziel prüfen
SERVFAIL nur bei validierendem Resolver	DNSSEC möglich
Name mit Punkt funktioniert, Kurzname nicht	Suchsuffix oder Suchreihenfolge

38. Welche typischen Fehlinterpretationen müssen vermieden werden?

Beobachtung	Falscher Schluss	Richtige Einordnung
Ping per IP funktioniert	DNS ist sicher die einzige Ursache	Dienst kann Hostnamen oder TLS benötigen
nslookup funktioniert	Anwendung verwendet dieselbe Auflösung	Anwendung kann eigenen Resolver verwenden
DNS-Server ist eingetragen	er wird tatsächlich verwendet	schnittstellen- und anwendungsbezogene Resolver prüfen

Beobachtung	Falscher Schluss	Richtige Einordnung
NXDOMAIN	DNS-Server ist ausgefallen	Server antwortet, Name existiert laut Antwort nicht
SERVFAIL	Name existiert nicht	Server konnte keine gültige Antwort liefern
NOERROR	gesuchter Record ist vorhanden	Antwort kann leer sein
DNS-Cache geleert und Zugriff geht	Ursache ist behoben	veraltete Quelle kann erneut cachen
A-Record ist korrekt	Dienst ist vollständig korrekt	AAAA, CNAME, TLS und Backend prüfen
öffentliche Antwort unterscheidet sich	eine Seite ist falsch	Split DNS kann beabsichtigt sein
Port 53 per TCP erreichbar	DNS funktioniert	UDP und tatsächliche Abfrage separat prüfen
keine Port-53-Pakete sichtbar	Anwendung stellt keine DNS-Anfrage	DoH, Cache oder falscher Messpunkt möglich
kurze Query Time	gesamte Anwendung ist schnell	nur DNS-Abfragedauer gemessen

39. Wie sieht der systematische Prüfablauf aus?

Schritt	Prüfung
1	betroffenen Namen und Recordtyp eindeutig bestimmen
2	Clientzeit, Netzwerk und VPN-Zustand dokumentieren
3	konfigurierte und tatsächlich verwendete Resolver bestimmen
4	Hosts-Datei und Suchsuffixe prüfen
5	Systemresolver abfragen
6	konfigurierten DNS-Server direkt abfragen
7	A, AAAA und mögliche CNAME-Kette untersuchen
8	Antwortstatus, TTL, Flags und Server dokumentieren
9	funktionierenden Client oder Resolver vergleichen
10	UDP- und TCP-Abfrage vergleichen
11	autoritative Server direkt prüfen
12	SOA-Seriennummern und Zonendaten vergleichen
13	Split DNS, VPN, DoH und Conditional Forwarding berücksichtigen

Schritt	Prüfung
14	bei Bedarf autorisierten Paketmitschnitt erstellen
15	Cache erst nach Beweissicherung kontrolliert leeren
16	DNS-Antwort und tatsächliche Dienstverbindung erneut testen
17	Ursache und Änderung dokumentieren

Merksatz

> Name → Recordtyp → Systemresolver → konfigurierter Resolver
> → autoritativer Server → Cache → tatsächliche Anwendung

40. Kompakte Befehlsübersicht für Windows, Linux und macOS

Aufgabe	Windows	Linux	macOS
DNS-Konfiguration	`[RO] Get-DnsClientServerAddress`	`[RO] resolvectl status`	`[RO][SENS] scutil --dns`
DNS-Suffixe	`[RO] Get-DnsClient`	`[RO] resolvectl status`	`[RO] networksetup -getsearchdomains "DIENST"`
vollständige Netzkonfiguration	`[RO][SENS] ipconfig /all`	`[RO] cat /etc/resolv.conf`	`[RO] networksetup -getinfo "DIENST"`
Systemauflösung	`[TEST] Resolve-DnsName NAME`	`[TEST] resolvectl query NAME`	`[TEST] dscacheutil -q host -a name NAME`
A-Record	`[TEST] Resolve-DnsName NAME -Type A`	`[TEST] dig NAME A`	`[TEST] dig NAME A`
AAAA-Record	`[TEST] Resolve-DnsName NAME -Type AAAA`	`[TEST] dig NAME AAAA`	`[TEST] dig NAME AAAA`
bestimmten Resolver abfragen	`[TEST] Resolve-DnsName NAME -Server DNS-SERVER`	`[TEST] dig @DNS-SERVER NAME`	`[TEST] dig @DNS-SERVER NAME`
TCP erzwingen	`[TEST] Resolve-DnsName NAME -Server DNS-SERVER -TcpOnly`	`[TEST] dig +tcp @DNS-SERVER NAME`	`[TEST] dig +tcp @DNS-SERVER NAME`
Reverse DNS	`[TEST] Resolve-DnsName IP -Type PTR`	`[TEST] dig -x IP @DNS-SERVER`	`[TEST] dig -x IP @DNS-SERVER`
NS-Records	`[TEST] Resolve-DnsName ZONE -Type NS`	`[TEST] dig @DNS-SERVER ZONE NS`	gleicher Befehl
SOA-Record	`[TEST] Resolve-DnsName ZONE -Type SOA`	`[TEST] dig @DNS-SERVER ZONE SOA`	gleicher Befehl
Hosts-Datei	`[RO][PRIV][SENS] Get-Content "$env:SystemRoot\System32\drivers\etc\hosts"`	`[RO][SENS] cat /etc/hosts`	`[RO][SENS] cat /etc/hosts`

Aufgabe	Windows	Linux	macOS
DNS-Cache anzeigen	`[RO] Get-DnsClientCache`	`[RO] resolvectl statistics`	kein gleichwertiger vollständiger Standardbefehl
DNS-Cache leeren	`[PRIV][CHANGE] Clear-DnsClientCache`	`[PRIV][CHANGE] sudo resolvectl flush-caches`	`[PRIV][CHANGE] sudo dscacheutil -flushcache`
DNS mitschneiden	`[TEST][PRIV][FILE][SENS] dumpcap -i INTERFACE -f "host DNS-SERVER and port 53" -a duration:30 -w dns.pcapng`	gleicher Befehl	gleicher Befehl

Die Platzhalter `NAME`, `ZONE`, `IP`, `DNS-SERVER`, `DIENST` und `INTERFACE` müssen durch vorher eindeutig bestimmte Werte ersetzt werden.

41. Dokumentationsvorlage

Ticketnummer:

Prüfzeitpunkt:

Zeitzone:

Betroffener Client:

Betriebssystem:

Benutzer:

Standort:

VLAN:

VPN aktiv:

Betroffene Anwendung:

ABFRAGE

Eingegebener Name:

Vollständig qualifizierter Name:

Recordtyp:

Erwartete Antwort:

Tatsächliche Antwort:

Exakte Fehlermeldung:

CLIENTKONFIGURATION

Schnittstelle:

IPv4-Adresse:

IPv6-Adresse:

DNS-Server IPv4:

DNS-Server IPv6:

DNS-Suchdomänen:

Lokale Hosts-Datei geprüft:

Passender Hosts-Eintrag:

DoH aktiv oder vermutet:

Proxy aktiv:

Anwendungscache:

SYSTEMRESOLVER

Verwendeter Test:

Antwortstatus:

Antwortrecords:

TTL:

CNAME-Kette:

Antwortzeit:

Verwendeter Resolver:

IPv4 oder IPv6:

DIREKTE DNS-ABFRAGE

DNS-Server:

UDP erfolgreich:

TCP erfolgreich:

A-Antwort:

AAAA-Antwort:

CNAME:

Status:

Flags:

TTL:

Query Time:

AUTORITATIVE DATEN

Zone:

Autoritative Nameserver:

Direkt abgefragter Server:

AA-Flag:

A-Record:

AAAA-Record:

CNAME:

SOA-Seriennummer:

Weitere autoritative Server:

Seriennummern identisch:

Delegation korrekt:

DNSSEC-Status:

VERGLEICH

Referenzclient:

Verwendeter Resolver:

Antwort:

TTL:

Anwendung funktioniert:

Relevante Abweichungen:

PAKETMITSCHNITT

Messpunkt:

DNS-Anfrage sichtbar:

DNS-Antwort sichtbar:

Transaktions-ID:

UDP oder TCP:

Response Code:

Truncated:

Wiederholungen:

Antwortzeit:

Capture-Datei:

Hash:

CACHE

Clientcache vor Änderung gesichert:

Resolvercache vermutet:

Negative Antwort:

Cache geleert:

Ergebnis nach Cacheleerung:

Arbeitshypothese:

Begründung:

Durchgeführter Test:

Geplante Änderung:

Freigabe:

Rückweg:

Abschlussprüfung:

Ermittelte Ursache:

42. Offizielle Quellen und weiterführende Dokumentation

Microsoft

- DNS – technische Übersicht
- Resolve-DnsName
- Get-DnsClientServerAddress
- Get-DnsClient
- Get-DnsClientCache
- Clear-DnsClientCache
- nslookup

Linux und systemd

- resolvectl
- systemd-resolved
- resolv.conf – Linux Manual Page

ISC BIND

- BIND 9 Administrator Reference Manual
- dig Manual

Apple

- Lokale Befehlsreferenzen: `man scutil`, `man networksetup`, `man dscacheutil`, `man dig` und `man nslookup`

- DNS-Einstellungen auf dem Mac ändern

Wireshark

- DNS Display Filter Reference
- dumpcap Manual Page
- tshark Manual Page

Standards

- RFC 1034 – Domain Names: Concepts and Facilities
- RFC 1035 – Domain Names: Implementation and Specification
- RFC 2308 – Negative Caching of DNS Queries
- RFC 6891 – Extension Mechanisms for DNS
- RFC 7766 – DNS Transport over TCP
- RFC 7858 – DNS over TLS
- RFC 8484 – DNS over HTTPS

> DNS-Verwaltungsbefehle und Serverprotokolle unterscheiden sich je nach Produkt. Änderungen an Zonen, Delegationen, Forwardern, DNSSEC oder Resolverrichtlinien dürfen nur anhand der Dokumentation des tatsächlich eingesetzten DNS-Systems vorgenommen werden.
