

[zurück](#)

# DNSSEC - Grundlagen

DNSSEC (**DNS Security Extensions**) erweitert das Domain Name System um Sicherheitsfunktionen. Es sorgt dafür, dass DNS-Antworten **authentisch**, **unverändert** und **vertrauenswürdig** sind.

Wichtig:

- DNSSEC schützt **NICHT** die Vertraulichkeit (keine Verschlüsselung),
- sondern die **Integrität** und **Authentizität** von DNS-Daten.

## Warum DNSSEC?

DNS selbst ist über 40 Jahre alt und ursprünglich ungeschützt. Ohne DNSSEC sind folgende Angriffe möglich:

- DNS-Spoofing
- Cache Poisoning
- Man-in-the-Middle
- Umleitung auf falsche Server

Beispiel: Ein Angreifer könnte Google.com auf eine gefälschte IP zeigen lassen.

DNSSEC verhindert genau diese Manipulationen.

## Wie funktioniert DNSSEC?

DNSSEC nutzt **digitale Signaturen**.

Jede Zone unterschreibt ihre DNS-Daten mit einem privaten Schlüssel.

Clients können prüfen:

- „Kommt diese DNS-Antwort wirklich vom Zonenbetreiber?“
- „Wurde sie unterwegs verändert?“

## Schlüsseltypen

DNSSEC arbeitet mit zwei Schlüsselarten:

### 1. ZSK - Zone Signing Key

- signiert die Resource Records in einer Zone
- wird häufiger gewechselt

## 2. KSK - Key Signing Key

- signiert den öffentlichen ZSK
- wird selten gewechselt
- bildet die Vertrauenskette (Chain of Trust)

## Wichtige DNSSEC Resource Records

RR-Typ	Bedeutung
-----	-----
<b>RRSIG</b>	digitale Signatur eines DNS-Eintrags
<b>DNSKEY</b>	öffentlicher Schlüssel
<b>DS</b>	Delegation Signer (Verweis auf Kindzone)
<b>NSEC / NSEC3</b>	beweist sicher „Eintrag existiert NICHT“

Beispiel RRSIG:

```
example.com. IN RRSIG A 8 2 3600 20250101000000 (...)
```

## Chain of Trust (Vertrauenskette)

DNSSEC baut eine Kette vom **Root-Nameserver** bis zur **Domain** auf.

ASCII-Visualisierung:

```
[ Root (.) ]
  ↓ DS
[ .com ]
  ↓ DS
[ example.com ]
  ↓ RRSIG
[ host.example.com ]
```

Jede Stufe bestätigt die nächste.

## Beispiel Ablauf einer DNSSEC-Prüfung

1. Client fragt eine Domain z. B. `example.com`.
2. Nameserver liefert Antwort + **RRSIG**.
3. Client lädt **DNSKEY** der Zone.
4. DNSKEY wird per **DS-Eintrag** in der Parent-Zone bestätigt.
5. Alles beginnt beim **Root**, das öffentlich bekannt ist.
6. Ergebnis:

- \* „Signatur gültig“ → Antwort akzeptiert
- \* „Signatur ungültig“ → Antwort wird verworfen

## Negative Antworten: NSEC / NSEC3

DNSSEC kann beweisen, dass ein Eintrag **wirklich nicht existiert**.

Beispiel: Du fragst nach `abc123.example.com`, der nicht existiert.

Ohne DNSSEC:

- Angreifer könnte beliebige Antwort fälschen

Mit DNSSEC:

- NSEC oder NSEC3 signiert die „Nicht-Existenz“

## Vorteile von DNSSEC

- Schutz vor Cache Poisoning
  - Schutz vor gefälschten DNS-Antworten
    - \* garantierte Datenintegrität
    - \* Basis für DANE (TLSA-Records für E-Mail-Sicherheit)

## Nachteile / Herausforderungen

- höherer Administrationsaufwand
  - größere DNS-Pakete
    - \* nicht alle Resolver unterstützen DNSSEC
    - \* Signaturen müssen regelmäßig erneuert werden

## Beispiel - DNSSEC aktiv prüfen

Linux:

```
dig example.com +dnssec
```

Gültige Antwort zeigt:

- DNSKEY
  - \* RRSIG
  - \* AD-Flag („Authenticated Data“)

## Zusammenfassung

- DNSSEC schützt **DNS-Integrität**, nicht Vertraulichkeit

- verwendet digitale Signaturen mit ZSK und KSK
  - \* Chain of Trust: Root → TLD → Domain
  - \* RRSIG, DNSKEY, DS, NSEC/NSEC3 sind die wichtigsten Recordtypen
  - \* verhindert Spoofing, MITM und Cache Poisoning

From: <http://wiki.nctl.de/dokuwiki/> - ☐ **Veni. Vidi. sudo rm -rf / vici.**

Permanent link: <http://wiki.nctl.de/dokuwiki/doku.php?id=it-themen:grundlagen:netzwerkdienste:dnssec&rev=1764585331>

Last update: **01.12.2025 11:35**

