

DNS und TLS

Einträge verstehen, Zertifikate richtig einsetzen.

- [DNS-Einträge erklärt](#)
- [TLS-Zertifikate: Kette, Ablauf, typische Fehler](#)

DNS-Einträge erklärt

Prüfen kannst du jeden Eintrag mit der [DNS-Abfrage](#).

Typ	Wofür	Beispiel
A	Name → IPv4-Adresse	www.example.de → 203.0.113.10
AAAA	Name → IPv6-Adresse	www.example.de → 2001:db8::10
CNAME	Name ist ein Alias für einen anderen Namen	shop.example.de → shops.anbieter.net
MX	Wer nimmt Mails für die Domain an, mit Priorität	10 mail.example.de
TXT	freier Text, z. B. SPF, DKIM, DMARC, Domain-Bestätigungen	"v=spf1 mx -all"
NS	Welche Nameserver für die Zone zuständig sind	ns1.provider.de
PTR	IP → Name (Reverse DNS), wird beim Inhaber der IP gesetzt	10.113.0.203.in-addr.arpa → mail.example.de
CAA	Welche Zertifizierungsstellen Zertifikate ausstellen dürfen	0 issue "letsencrypt.org"

Regeln, die oft vergessen werden

- **CNAME steht allein:** Ein Name mit CNAME darf keine anderen Einträge haben. Deshalb geht CNAME nicht auf der Domain selbst (example.de), dort stehen ja MX und NS.
- **MX zeigt auf einen Namen, nie auf eine IP**, und dieser Name braucht einen A/AAAA-Eintrag, kein CNAME.
- **TTL:** So lange speichern Resolver die Antwort zwischen. Vor einem Umzug die TTL ein paar Tage vorher auf 300 Sekunden senken, danach wieder hoch.
- **Reverse DNS (PTR)** setzt nicht deine DNS-Zone, sondern der Inhaber der IP, also dein Hoster oder Provider.

Änderung kommt nicht an?

In der DNS-Abfrage siehst du Resolver und zuständige Nameserver nebeneinander:

- **Nameserver alt, Resolver alt:** Die Änderung ist noch nicht in der Zone. Beim DNS-Anbieter prüfen.
- **Nameserver neu, Resolver alt:** alles richtig, warte die TTL ab.
- **Nameserver uneinig:** Ein Nameserver ist nicht synchron, oder beim Registrar stehen falsche NS-Einträge.

TLS-Zertifikate: Kette, Ablauf, typische Fehler

Prüfen kannst du jeden Server mit dem [SSL/TLS-Check](#), auch Mailserver per STARTTLS.

Die Zertifikatskette

Ein Browser vertraut einer kleinen Zahl von **Wurzelzertifikaten**. Dein Server-Zertifikat ist aber von einem **Zwischenzertifikat** signiert. Damit der Client die Kette prüfen kann, muss der Server ihm **sein Zertifikat und alle Zwischenzertifikate** schicken, die Wurzel nicht.

Häufigster Fehler: Es wird nur das eigene Zertifikat ausgeliefert. Browser holen das Fehlende manchmal selbst nach, Apps, Skripte und Mailserver nicht. Die Folge: „unable to get local issuer certificate“.

Software	richtig
nginx	<code>ssl_certificate /etc/letsencrypt/live/example.de/fullchain.pem;</code>
Apache 2.4.8+	<code>SSLCertificateFile ../fullchain.pem</code>
Postfix	<code>smtpd_tls_chain_files = ../privkey.pem, ../fullchain.pem</code>

Ablauf und Erneuerung

Zertifikate von Let's Encrypt gelten 90 Tage, künftig noch kürzer. Ohne automatische Erneuerung läuft es garantiert irgendwann ab.

- **certbot:** `certbot renew --dry-run` testet die Erneuerung. Nach dem Erneuern muss der Dienst neu laden, z. B. mit `--deploy-hook "systemctl reload nginx"`.
- **Caddy, Traefik** erneuern selbst.
- **Nicht vergessen:** Mailserver, VPN, Proxmox, NAS und alles, was ein Zertifikat per Hand bekommen hat.

Damit es nicht überraschend kommt: Im [Monitoring](#) (Premium) eine Zertifikats-Prüfung anlegen, sie warnt 14 Tage vorher.

Typische Fehlermeldungen

Meldung	Ursache
Name passt nicht (<code>hostname mismatch</code>)	Der aufgerufene Name steht nicht in den SANs des Zertifikats. Mit allen Namen neu ausstellen.
Zertifikat abgelaufen	Erneuerung läuft nicht oder der Dienst hat das neue Zertifikat nicht geladen
selbst signiert	Kein Zertifikat einer öffentlichen Stelle. Für öffentliche Dienste Let's Encrypt nehmen.
unvollständige Kette	siehe oben, <code>fullchain</code> statt <code>cert</code>

Protokolle und HSTS

- **TLS 1.2 und 1.3 an, 1.0 und 1.1 aus.** Alte Versionen braucht heute kein aktueller Client mehr.
- **HSTS** (`Strict-Transport-Security: max-age=31536000`) sorgt dafür, dass Browser nur noch per HTTPS kommen. Erst setzen, wenn HTTPS überall sicher läuft, auch auf Subdomains, falls du `includeSubDomains` nutzt.