x.509 Zertifikate für https und andere Anwendungsfälle

Linux-Cafe 2026-11-11

Bernd Strößenreuther

Lizenz

Sie dürfen die Text-Inhalte dieses Dokument verwenden unter den Bedingungen der Creative Commons Lizenz:

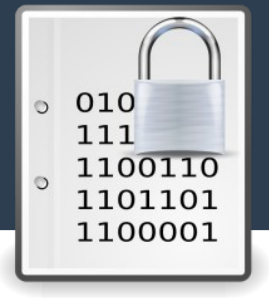
http://creativecommons.org/licenses/by-nc-sa/3.0/de/

Alle nicht separat gekennzeichnenten Grafiken und Icons von OpenClipArt.org / freesvg.org - ”released to the public domain”.

Agenda

- **Häufigster Anwendungsfall: https**
- **Begriffsklärung**
  - Zertifikat / x.509 / TLS / mTLS / openssl / CA / PKI / …
- **ACME**
- **Demo: Let’s Encrypt / CertBot**
- **Maximale Gültigkeit von TLS-Zertifikaten**
- **Welche CA für welchen Anwendungsfall?**
- **Fragen**

https



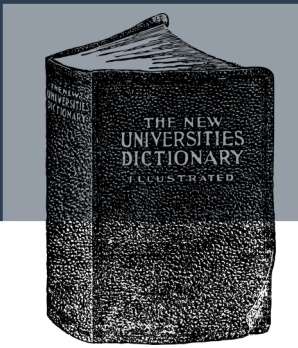
- **Was soll erreicht werden, mit dem Einsatz von https?**
  - Verschlüsselung der Verbindung (Vertraulichkeit)
  - Integrität der Daten gewährleisten
  - Identität des Servers nachweisen
  - ggf. Identität des Clients bzw. Users nachweisen
- **Was braucht man dazu?**
  - (mindestens) ein Schlüsselpaar: Private + Public Key
  - Digitale Signatur des Public Key, ausgestellt von einer vertauenswürdigen Stelle: Das Zertifikat

kurze Werbepause



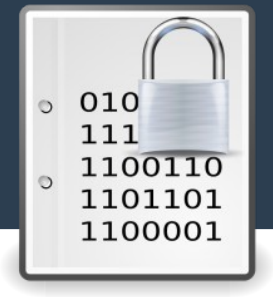
- **TLS4Developers Workshop: Hands-on-Labs**
  https://github.com/booboo-at-gluga-de/TLS4Developers_Workshop

Begriffsklärung



- **CA:** Certificate Authority – vertrauenswürdige Stelle, Root-Zertifikat
- **PKI:** Public Key Infrastructure – Gesamtlösung, Source of Trust
- **CSR:** Certificate Signing Request – lokal erzeugen, an die CA schicken
- **Zertifikat:** Public Key + Signatur – kommt von der CA zurück
- **x.509:** Standard, der definiert wie Zertifikate, Keys, etc. aussehen
- **TLS (aka. Standard-TLS):** Nur der Server weist sich aus
- **mTLS:** Auch der Client weist sich aus (mit einem Client-Zertifikat)
- **openssl:** implementiert x.509 - eine Lib und ein Commandline Tool

TLS – Transport Layer Security

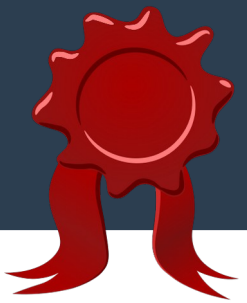


- **Nur die Verbindung wird verschlüsselt!**
  - Auf der (physikalischen) Leitung fließen verschlüsselte Daten
  - innerhalb der TLS-Verbindung fließen Klartext-Daten(!!)
- **Keine End-2-End Encryption!**
- **Einsatz:** https, imaps, smtps, DNS over TLS, etc.

TLS-Versionen

- **TLS 1.3**
  - die aktuelle Version (von 2018)
  - unbedingt nutzen, wo auch immer möglich
    (Post Quantum Algorithmen gibt es nur dafür)
- **TLS 1.2**
  - kann eine Alternative sein, wenn sehr alte Clients oder Server zwingend noch unterstützt werden müssen
- **TLS 1.1 / TLS 1.0 / SSL 3.0 / SSL 2.0 / SSL 1.0**
  - Finger weg! Auf allen Clients und allen Servern hart abschalten!

Intermediate Zertifikate



- **Root-Zertifikate sind extrem weit verbreitet**
  auf mehr oder weniger jedem Device weltweit in mindestens einem Truststore: Austausch bei Kompromittierung praktisch unmöglich
- **bei den offiziellen CAs: Immer Offline**
- **Zwischengeschaltet immer (mindestens) eine Intermediate-CA**
  die Online ist (dafür aber revoked werden könnte)
- **Aufgabenverteilung:** Wer das End-Entity Zertifikat hält, liefert im SSL-Handshake auch alle erforderlichen Intermediate-Zertifikate mit!
- **Sicherheitshinweis:** Packt niemals ein Intermediate-Zertifikat in einen Truststore – sonst funktioniert die Revocation nicht mehr!

Let’s Encrypt / ACME

Automatic Certificate Management Environment (ACME)

RFC 8555

Voraussetzungen:

- **Ein Domain Name, der öffentlich im DNS auflösbar ist**
- **Nachweis, dass man „Kontrolle“ über die Domain hat**
- **Ein ACME-Client**

Nicht erforderlich:

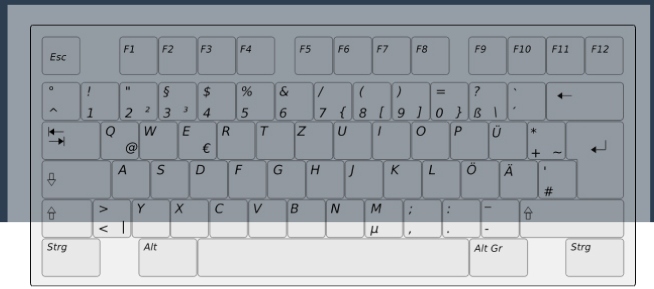
- **Der Server muss nicht (dauerhaft) von außen erreichbar sein**

ACME-Challenges

Nachweis, dass man „Kontrolle“ über die Domain hat

- **https://letsencrypt.org/docs/challenge-types/**
- **HTTP-01 challenge**
  - Bestimmten Inhalt auf Port 80 per HTTP zur Verfügung stellen
- **DNS-01 challenge**
  - Bestimmten Inhalt in einem TXT record unter der Domain im DNS zur Verfügung stellen

Demo-Time



- **Let’s Encrypt / CertBot**

Maximale Gültigkeit von TLS-Zertifikaten

- **Certificate Revocation sucks!**
- **Daher beschloss das CA/Browser Forum Anfang 2025 in Ballot SC-081v3 die stufenweise Verkürzung des maximalen Gültigkeitszeitraums von TLS-Zertifikaten**
  - bis 2026-03-15: 398 Tage
  - ab 2026-03-15: 200 Tage
  - ab 2027-03-15: 100 Tage
  - ab 2029-03-15: 47 Tage
- **nachzulesen unter**
  https://www.digicert.com/blog/tls-certificate-lifetimes-will-officially-reduce-to-47-days

Welche CA für welchen Anwendungsfall?

Server-Zertifikate

- **Sicherstellen:** Spreche ich mit dem echten Server?
- **Szenario:** Viele Clients

Client-Zertifikate

- **Sicherstellen:** Authentifikation - spricht da der echte Client bzw. User?
- **Szenario:** Wenige Server

Zertifikate einer offiziellen CA klar im Vorteil

Abwägung: Eigene CA besser geeignet?


Noch Fragen?