OpenSSL Befehle

Zertifikate prüfen, Ketten debuggen, Formate umwandeln – die openssl Befehle für den Alltag, durchsuchbar und mit einem Klick kopierbar.

daniele.michelin // cheat sheet

Verbindung testen

TLS-Verbindung mit SNI aufbauen und Handshake anzeigen

```
openssl s_client -connect example.com:443 -servername example.com </dev/null
```

Komplette ausgelieferte Zertifikatskette anzeigen

```
openssl s_client -connect example.com:443 -servername example.com -showcerts </dev/null
```

Gültigkeitszeitraum des Server-Zertifikats

```
openssl s_client -connect example.com:443 -servername example.com </dev/null 2>/dev/null | openssl x509 -noout -dates
```

Subject, Aussteller und SANs des Servers

```
openssl s_client -connect example.com:443 -servername example.com </dev/null 2>/dev/null | openssl x509 -noout -subject -issuer -ext subjectAltName
```

OCSP-Stapling prüfen

```
openssl s_client -connect example.com:443 -servername example.com -status </dev/null 2>/dev/null | grep -A 3 'OCSP Response'
```

STARTTLS auf SMTP-Submission testen

```
openssl s_client -starttls smtp -connect mail.example.com:587 -servername mail.example.com </dev/null
```

STARTTLS auf IMAP testen

```
openssl s_client -starttls imap -connect mail.example.com:143 </dev/null
```

IMAPS (implizites TLS) testen

```
openssl s_client -connect mail.example.com:993 -servername mail.example.com </dev/null
```

TLS-Versionen & Cipher

Prüfen, ob TLS 1.3 angeboten wird

```
openssl s_client -connect example.com:443 -servername example.com -tls1_3 </dev/null
```

Prüfen, ob TLS 1.2 angeboten wird

```
openssl s_client -connect example.com:443 -servername example.com -tls1_2 </dev/null
```

Veraltetes TLS 1.1 testen (sollte fehlschlagen)

```
openssl s_client -connect example.com:443 -servername example.com -tls1_1 </dev/null
```

Bestimmte Cipher-Suite erzwingen

```
openssl s_client -connect example.com:443 -servername example.com -tls1_2 -cipher 'ECDHE-RSA-AES256-GCM-SHA384' </dev/null
```

Cipher auflisten, die ein String freigibt

```
openssl ciphers -v 'HIGH:!aNULL:!MD5'
```

Zertifikate prüfen

Zertifikat vollständig lesbar anzeigen

```
openssl x509 -in cert.pem -noout -text
```

Ablaufdatum anzeigen

```
openssl x509 -in cert.pem -noout -enddate
```

Läuft in 30 Tagen ab? (Exit-Code 1 = ja)

```
openssl x509 -in cert.pem -noout -checkend 2592000
```

SHA-256-Fingerprint anzeigen

```
openssl x509 -in cert.pem -noout -fingerprint -sha256
```

Kette gegen Root und Intermediate prüfen

```
openssl verify -CAfile root.pem -untrusted intermediate.pem cert.pem
```

Keys & CSR

Key und CSR mit SANs in einem Schritt erzeugen

```
openssl req -new -newkey rsa:4096 -nodes -keyout example.key -out example.csr -subj "/CN=example.com" -addext "subjectAltName=DNS:example.com,DNS:www.example.com"
```

ECDSA-Key (P-256) erzeugen

```
openssl genpkey -algorithm EC -pkeyopt ec_paramgen_curve:P-256 -out example.key
```

CSR anzeigen und Signatur prüfen

```
openssl req -in example.csr -noout -text -verify
```

Selbstsigniertes Testzertifikat erstellen

```
openssl req -x509 -newkey rsa:2048 -nodes -days 365 -keyout test.key -out test.crt -subj "/CN=test.local"
```

Passen Key und Zertifikat? Hashes müssen gleich sein

```
openssl x509 -in cert.pem -noout -pubkey | openssl sha256; openssl pkey -in key.pem -pubout | openssl sha256
```

RSA-Abgleich über Modulus (klassische Methode)

```
openssl x509 -in cert.pem -noout -modulus | openssl md5; openssl rsa -in key.pem -noout -modulus | openssl md5
```

Formate umwandeln

PEM nach DER umwandeln

```
openssl x509 -in cert.pem -outform der -out cert.der
```

DER nach PEM umwandeln

```
openssl x509 -in cert.der -inform der -out cert.pem
```

PFX/P12 aus Key, Zertifikat und Kette bauen (Windows/IIS)

```
openssl pkcs12 -export -out cert.pfx -inkey key.pem -in cert.pem -certfile chain.pem
```

! PFX entpacken – Key liegt danach unverschlüsselt vor

```
openssl pkcs12 -in cert.pfx -nodes -out all.pem
```

! Passphrase von einem Key entfernen

```
openssl pkey -in encrypted.key -out plain.key
```

Hilfsfunktionen

Zufälliges Passwort erzeugen (32 Zeichen)

```
openssl rand -base64 24
```

Zufälliges Secret als Hex (z. B. für Tokens)

```
openssl rand -hex 32
```

SHA-256-Prüfsumme einer Datei

```
openssl dgst -sha256 file.iso
```

OpenSSL-Version und Build-Optionen anzeigen

```
openssl version -a
```