OpenSSL cheat sheet

Check certificates, debug chains, convert formats – everyday openssl commands, searchable and copyable in one click.

daniele.michelin // cheat sheet

Test connections

Connect with SNI and show the TLS handshake

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

Show the full certificate chain served

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

Validity period of the server certificate

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

Subject, issuer and SANs of the server cert

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

Check OCSP stapling

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

Test STARTTLS on SMTP submission

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

Test STARTTLS on IMAP

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

Test IMAPS (implicit TLS)

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

TLS versions & ciphers

Check whether TLS 1.3 is offered

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

Check whether TLS 1.2 is offered

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

Test legacy TLS 1.1 (should fail)

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

Force a specific cipher suite

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

List ciphers matched by a cipher string

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

Inspect certificates

Show a certificate in full, human-readable

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

Show the expiry date

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

Expires within 30 days? (exit code 1 = yes)

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

Show the SHA-256 fingerprint

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

Verify the chain against root and intermediate

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

Keys & CSR

Create key and CSR with SANs in one step

```
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"
```

Generate an ECDSA key (P-256)

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

Show a CSR and verify its signature

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

Create a self-signed test certificate

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

Do key and cert match? Hashes must be equal

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

RSA match via modulus (classic method)

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

Convert formats

Convert PEM to DER

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

Convert DER to PEM

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

Build a PFX/P12 from key, cert and chain (Windows/IIS)

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

! Extract a PFX – the key ends up unencrypted

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

! Remove the passphrase from a key

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

Utilities

Generate a random password (32 chars)

```
openssl rand -base64 24
```

Random secret as hex (e.g. for tokens)

```
openssl rand -hex 32
```

SHA-256 checksum of a file

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

Show OpenSSL version and build options

```
openssl version -a
```