ssl, certificate, certificatechain, chain, gpg, tor, torproject, monit, howto, certutil, asc, validate, verify

check SSL certificates and certificate chains

- https://www.sslshopper.com/article-most-common-openssl-commands.html
- https://superuser.com/questions/806743/mozilla-nss-certutil-binary
- https://unix.stackexchange.com/questions/368123/how-to-extract-the-root-ca-and-subordinate-ca-from-a-certificate-chain-in-linux
- **WINDOWS-SPECIFIC!**: https://sysadmins.lv/retired-msft-blogs/pki/basic-crl-checking-with-certutil.aspx

certutil

- https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS/tools/NSS_Tools_certutil

Verify tor project signing keys at https://2019.www.torproject.org/docs/signing-keys.html.en . Get latest release of mar-tools-linux64.zip from tor project at https://dist.torproject.org/torbrowser/<VERSION>/mar-tools-linux64.zip .

```
mkdir -pv ~/Downloads/mar-tools
wget -O ~/Downloads/mar-tools/mar-tools-linux64.zip
https://dist.torproject.org/torbrowser/10.5a8/mar-tools-linux64.zip
wget -O ~/Downloads/mar-tools/mar-tools-linux64.zip.asc
https://dist.torproject.org/torbrowser/10.5a8/mar-tools-linux64.zip.asc
gpg --verify --auto-key-retrieve ~/Downloads/mar-tools/mar-tools-linux64.zip.asc ~/Downloads/mar-
tools/mar-tools-linux64.zip
```

```
gpg: Signature made Tue 26 Jan 2021 08:59:37 AM CET
gpg:                using RSA key EB774491D9FF06E2
gpg: key 4E2C6E8793298290: public key "Tor Browser Developers (signing key) <torbrowser@torproject.org>"
imported
gpg: Total number processed: 1
gpg:               imported: 1
gpg: Good signature from "Tor Browser Developers (signing key) <torbrowser@torproject.org>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: EF6E 286D DA85 EA2A 4BA7  DE68 4E2C 6E87 9329 8290
     Subkey fingerprint: 1107 75B5 D101 FB36 BC6C  911B EB77 4491 D9FF 06E2
```

```
unzip -d ~/Downloads/mar-tools/ ~/Downloads/mar-tools/mar-tools-linux64.zip
~/Downloads/mar-tools/mar-tools/certutil --syntax
```

```
Type certutil -H for more detailed descriptions
Usage:  certutil -N [-d certdir] [-P dbprefix] [-f pwfile] [--empty-password]
Usage:  certutil -T [-d certdir] [-P dbprefix] [-h token-name]
                 [-f pwfile] [-0 SSO-password]
        certutil -A -n cert-name -t trustargs [-d certdir] [-P dbprefix] [-a] [-i input]
        certutil -B -i batch-file
        certutil -C [-c issuer-name | -x] -i cert-request-file -o cert-file
                 [-m serial-number] [-w warp-months] [-v months-valid]
                 [-f pwfile] [-d certdir] [-P dbprefix] [-Z hashAlg]
                 [-1 | --keyUsage [keyUsageKeyword,..]] [-2] [-3] [-4]
                 [-5 | --nsCertType [nsCertTypeKeyword,...]]
                 [-6 | --extKeyUsage [extKeyUsageKeyword,...]] [-7 emailAddrs]
                 [-8 dns-names] [-a]
        certutil -D -n cert-name [-d certdir] [-P dbprefix]
        certutil --rename -n cert-name --new-n new-cert-name
                 [-d certdir] [-P dbprefix]
        certutil -E -n cert-name -t trustargs [-d certdir] [-P dbprefix] [-a] [-i input]
        certutil -F -n cert-name [-d certdir] [-P dbprefix]
        certutil -F -k key-id [-d certdir] [-P dbprefix]
        certutil -G -n key-name [-h token-name] [-k rsa] [-g key-size] [-y exp]
                 [-f pwfile] [-z noisefile] [-d certdir] [-P dbprefix]
        certutil -G [-h token-name] -k dsa [-q pqgfile -g key-size] [-f pwfile]
                 [-z noisefile] [-d certdir] [-P dbprefix]
```

```
        certutil -G [-h token-name] -k ec -q curve [-f pwfile]
                 [-z noisefile] [-d certdir] [-P dbprefix]
        certutil -K [-n key-name] [-h token-name] [-k dsa|ec|rsa|all]
                 [-f pwfile] [-X] [-d certdir] [-P dbprefix]
        certutil --upgrade-merge --source-dir upgradeDir --upgrade-id uniqueID
                 [--upgrade-token-name tokenName] [-d targetDBDir]
                 [-P targetDBPrefix] [--source-prefix upgradeDBPrefix]
                 [-f targetPWfile] [-@ upgradePWFile]
        certutil --merge --source-dir sourceDBDir [-d targetDBdir]
                 [-P targetDBPrefix] [--source-prefix sourceDBPrefix]
                 [-f targetPWfile] [-@ sourcePWFile]
        certutil -L [-n cert-name] [-h token-name] [--email email-address]
                 [-X] [-r] [-a] [--dump-ext-val OID] [-d certdir] [-P dbprefix]
        certutil --build-flags
        certutil -M -n cert-name -t trustargs [-d certdir] [-P dbprefix]
        certutil -O -n cert-name [-X] [-d certdir] [-a] [-P dbprefix]
                 [--simple-self-signed]
        certutil -R -s subj -o cert-request-file [-d certdir] [-P dbprefix] [-p phone] [-a]
                 [-7 emailAddrs] [-k key-type-or-id] [-h token-name] [-f pwfile]
                 [-g key-size] [-Z hashAlg]
        certutil -V -n cert-name -u usage [-b time] [-e] [-a]
                [-X] [-d certdir] [-P dbprefix]
Usage:  certutil -W [-d certdir] [-f pwfile] [-@newpwfile]
        certutil -S -n cert-name -s subj [-c issuer-name | -x]  -t trustargs
                 [-k key-type-or-id] [-q key-params] [-h token-name] [-g key-size]
                 [-m serial-number] [-w warp-months] [-v months-valid]
                 [-f pwfile] [-d certdir] [-P dbprefix] [-Z hashAlg]
                 [-p phone] [-1] [-2] [-3] [-4] [-5] [-6] [-7 emailAddrs]
                 [-8 DNS-names]
                 [--extAIA] [--extSIA] [--extCP] [--extPM] [--extPC] [--extIA]
                 [--extSKID] [--extNC] [--extSAN type:name[,type:name]...]
                 [--extGeneric OID:critical-flag:filename[,OID:critical-flag:filename]...]
        certutil -U [-X] [-d certdir] [-P dbprefix]
```

---

mit openssl die individuellen Komponenten einer Zertifikatskette verifizieren

- https://security.stackexchange.com/questions/118062/use-openssl-to-individually-verify-components-of-a-certificate-chain
- https://unix.stackexchange.com/questions/368123/how-to-extract-the-root-ca-and-subordinate-ca-from-a-certificate-chain-in-linux
- https://unix.stackexchange.com/questions/354195/download-and-verify-certificate-chain
- https://langui.sh/2009/03/14/checking-a-remote-certificate-chain-with-openssl/

In */etc/ssl/certs/* liegen die Zertifikate.

```
openssl s_client -showcerts -verify 5 -connect www.internetx.de:443 < /dev/null | awk '/BEGIN/,/END/{
if(/BEGIN/){a++}; out="cert"a".pem"; print >out}'
# Doesn't work in my case
# for cert in *.pem; do newname=$(openssl x509 -noout -subject -in $cert | sed -n 's/^.*CN=\(.*\)$/\1/;
s/[ ,.*]/_/g; s/__/_/g; s/^_//g;p').pem; mv $cert $newname; done
```

```
verify depth is 5
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2
verify return:1
depth=1 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust TLS RSA CA G1
verify return:1
depth=0 C = DE, ST = Bayern, L = Regensburg, O = InterNetX GmbH, CN = *.internetx.de
verify return:1
DONE
```

```
ls -la
```

```
total 7180
drwxr-xr-x 3 user group    4096 Feb  3 19:27 .
drwxr-xr-x 8 user group   12288 Feb  4 00:29 ..
-rw-r--r-- 1 user group    2663 Feb  4 01:57 cert1.pem
-rw-r--r-- 1 user group    1639 Feb  4 01:57 cert2.pem
-rw-r--r-- 1 user group    1294 Feb  4 01:57 cert3.pem
```

```
openssl verify -no-CApath -partial_chain -trusted cert2.pem cert1.pem
```

```
cert1.pem: OK
```

```
openssl verify -no-CApath -partial_chain -trusted cert3.pem cert2.pem
```

```
cert2.pem: OK
```

```
openssl verify -no-CApath -trusted /etc/ssl/certs/DigiCert_Global_Root_G2.pem cert3.pem
```

```
cert3.pem: OK
```

oder auch:

```
openssl verify -CApath /etc/ssl/certs cert3.pem
```

```
cert3.pem: OK
```

Zertifikatskette lesen

- https://security.stackexchange.com/questions/107240/how-to-read-certificate-chains-in-openssl
- https://serverfault.com/questions/589590/understanding-the-output-of-openssl-s-client

CAcert

```
openssl s_client -connect www.cacert.org:443 < /dev/null
```

```
CONNECTED(00000003)
depth=2 O = Root CA, OU = http://www.cacert.org, CN = CA Cert Signing Authority, emailAddress =
support@cacert.org
verify error:num=19:self signed certificate in certificate chain
verify return:1
depth=2 O = Root CA, OU = http://www.cacert.org, CN = CA Cert Signing Authority, emailAddress =
support@cacert.org
verify return:1
depth=1 O = CAcert Inc., OU = http://www.CAcert.org, CN = CAcert Class 3 Root
verify return:1
depth=0 C = AU, ST = NSW, L = Sydney, O = CAcert Inc., CN = www.cacert.org
verify return:1
---
Certificate chain
 0 s:C = AU, ST = NSW, L = Sydney, O = CAcert Inc., CN = www.cacert.org
   i:O = CAcert Inc., OU = http://www.CAcert.org, CN = CAcert Class 3 Root
 1 s:O = CAcert Inc., OU = http://www.CAcert.org, CN = CAcert Class 3 Root
   i:O = Root CA, OU = http://www.cacert.org, CN = CA Cert Signing Authority, emailAddress =
support@cacert.org
```

```
 2 s:O = Root CA, OU = http://www.cacert.org, CN = CA Cert Signing Authority, emailAddress =
support@cacert.org
   i:O = Root CA, OU = http://www.cacert.org, CN = CA Cert Signing Authority, emailAddress =
support@cacert.org
```

Certificate chain		
	Ausgabe	**Bedeutung**
0	s:C = AU, ST = NSW, L = Sydney, O = CAcert Inc., CN = www.cacert.org	Leaf-Zertifikat
	i:O = CAcert Inc., OU = http://www.CAcert.org, CN = CAcert Class 3 Root	Zertifikat mit dem das Leaf Zertifikat signiert wurde.
1	s:O = CAcert Inc., OU = http://www.CAcert.org, CN = CAcert Class 3 Root	Intermediate Zertifikat (chain Zertifikat)
	i:O = Root CA, OU = http://www.cacert.org, CN = CA Cert Signing Authority, emailAddress = support@cacert.org	Zertifikat mit dem das Intermediate zertifikat signiert wurde
2	s:O = Root CA, OU = http://www.cacert.org, CN = CA Cert Signing Authority, emailAddress = support@cacert.org	Das “Root-Zertifikat”, das verwendet wurde um das Intermediate Zertifikat zu signieren.
	i:O = Root CA, OU = http://www.cacert.org, CN = CA Cert Signing Authority, emailAddress = support@cacert.org	Das Root-Zertifikat wurde verwendet um das Root-Zertifikat zu signieren (deswegen die Meldung “verify error:num=19:self signed certificate in certificate chain ”)

InternetX

```
openssl s_client -connect www.internetx.de:443 < /dev/null
```

```
CONNECTED(00000003)
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2
verify return:1
depth=1 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust TLS RSA CA G1
verify return:1
depth=0 C = DE, ST = Bayern, L = Regensburg, O = InterNetX GmbH, CN = *.internetx.de
verify return:1
---
Certificate chain
 0 s:C = DE, ST = Bayern, L = Regensburg, O = InterNetX GmbH, CN = *.internetx.de
   i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust TLS RSA CA G1
 1 s:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust TLS RSA CA G1
   i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2
 2 s:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2
   i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2
```

Certificate chain		
	Ausgabe	**Bedeutung**
0	s:C = DE, ST = Bayern, L = Regensburg, O = InterNetX GmbH, CN = *.internetx.de	Leaf-Zertifikat (Wildcard)
	i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust TLS RSA CA G1	Zertifikat mit dem das Leaf Zertifikat signiert wurde.
1	s:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust TLS RSA CA G1	Intermediate Zertifikat (chain Zertifikat)
	i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2	Zertifikat mit dem das Intermediate zertifikat signiert wurde
2	s:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2	Das “Root-Zertifikat”, das verwendet wurde um das Intermediate Zertifikat zu signieren.
	i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2	Das Root-Zertifikat wurde verwendet um das Root-Zertifikat zu signieren (Das Zertifikat ist unter den lokalen Zertifikaten, denen vertraut wird (trusted))

Details anzeigen

```
openssl x509 -in /etc/ssl/certs/DigiCert_Global_Root_G2.pem -fingerprint -subject -issuer -serial -hash
-noout
```

```
SHA1 Fingerprint=DF:3C:24:F9:BF:D6:66:76:1B:26:80:73:FE:06:D1:CC:8D:4F:82:A4
```

```
subject=C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2
issuer=C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root G2
serial=033AF1E6A711A9A0BB2864B11D09FAE5
607986c7
```

Nur die Serial:

```
openssl x509 -in /etc/ssl/certs/DigiCert_Global_Root_G2.pem -fingerprint -subject -issuer -serial -hash
-noout | sed  -n /^[0-9]/p
```

TCP Port 443 (https) - Zugriff mit openssl prüfen

- https://www.thomas-krenn.com/de/wiki/TCP_Port_443_(https)_Zugriff_mit_openssl_%C3%BCberpr%C3%BCfen
- https://docs.pingidentity.com/bundle/solution-guides/page/iqs1569423823079.html

SSL and SSL Certificates Explained For beginners

- http://www.steves-internet-guide.com/ssl-certificates-explained/

~~DISCUSSION~~