Setting up Certifying Authority using OpenSSL

Create directory structure

Root

certs

crl

newcerts

private (chmod 700 private)

index.txt

serial (echo 1000 > serial)

openssl.cnf

intermediate

certs

crl

crs

newcerts

private (chod 700)

index.txt

serial (echo 1000 > serial)

crlnumber (echo 1000 > clrnumber)

1. Set up a Root Certifying Authority (RootCA). Create a public key-private key pair for RootCA and create a self-signed digital certificate for RootCA.

Self signed digital certificate (root ca)

* Copy contents to openssl.cnf and change dir path
* Create private key: “openssl genrsa –aes256 –out private/RootCA.key 4096”
* Change permission: “chmod 400 private/RootCA.key”
* Create root certificate: “openssl req –config openssl.cnf –key private/RootCA.key –new –x509 –days 7000 –sha256 –extensions v3\_ca –out RootCA.cert”
* Change RootCA.cert permission: “chmod 444 RootCA.cert”
* Move ‘RootCA.cert’ to certs subdirectory: “mv RootCA.cert certs”
* Check contents: “openssl x509 –noout –text –in certs/RootCA.cert”

1. Set up an Intermediate Certifying Authority (CSTCA). Create a public key-private key pair for CSTCA and get a digital certificate issued for it signed by RootCA.

digital certificate signed by RootCA

* Copy contents to openssl.cnf and change dir path
* Create private key: “openssl genrsa –aes256 –out intermediate/private/ CSTCA.key 4096”
* Change permission: “chmod 400 intermediate/private/ CSTCA.key”
* Create intermediate certificate request: “openssl req –config intermediate/openssl.cnf –new –sha256 –key intermediate/private/CSTCA.key –out intermediate/csr/CSTCA.csr”
* Create intermediate certificate: “openssl ca –config openssl.cnf –extensions v3\_intermediate\_ca -days 3650 –notext –md sha256 –in intermediate/csr/CSTCA.csr –out intermediate/certs/CSTCA.cert”
* Check certificate content: “openssl x509 –noout –text –in intermediate/certs/CST.cert”
* Verify certificate using root ca certificate : “openssl verify –CAfile certs/RootCA.cert intermediate/certs/CST.cert ”
* Create file to verify digital certificates issued by intermediate: “cat intermediate/certs/CST.cert certs/RooCA.cert > intermediate/certs/CA-Chain.cert”

1. Create a public–private key pair for website www.gbs.unigoa.ac.in and get is public key certified by CSTCA.

* Issuing certificate to end users: “openssl genrsa –aes256 –out intermediate/private/www.gbs.unigoa.ac.in.key 2048”
* “openssl req –config intermediate/openssl.cnf –key intermediate/private/www.gbs.unigoa.ac.in.key –new –sha256 –out intermediate/csr/www.gbs.unigoa.ac.in.csr”
* Move ‘CST.cert’ to certs subdirectory: “mv CSTCA.cert certs”
* “openssl ca –config intermediate/openssl.cnf –extensions server\_cert –days 375 –notext –md sha256 –in intermediate/csr/www.gbs.unigoa.ac.in.csr –out intermediate/csr/www.gbs.unigoa.ac.in.cert”

1. Demonstrate how certificates can be verified.

* “openssl verify –CAfile intermediate/certs/CSTCA.cert intermediate/certs/www.gbs.unigoa.ac.in.cert”
* “openssl verify –CAfile intermediate/certs/CSTCA.cert intermediate/certs/www.gbs.unigoa.ac.in.cert”

1. Create certificates for individual users and show how to revoke certificates. Create a Certificate Revocation List (CRL).

Create certificate for alice

* “openssl genrsa -aes256 –out intermediate/private/alice.key 2048”
* “openssl req –config intermediate/openssl.cnf –key intermediate/private/alice.key –new –sha256 –out intermediate/csr/alice.csr”
* “openssl ca –config intermediate/openssl.cnf –extensions user\_cert –days 375 –notext –md sha256 –in intermediate/csr/alice.csr –out intermediate/certs/alice.cert ”

Revoking

* “openssl ca –config intermediate/openssl.cnf –gencrl –out intermediate/crl/CSTCA.crl”
* “openssl crl –text –noout –in intermediate/crl/CSTCA.crl”

Revoking using openssl ca command

* “openssl ca –config intermediate/openssl.cnf –revoke intermediate/cerrts/dart.cert”
* “openssl ca –config intermediate/openssl.cnf –gencrl –out intermediate/crl/CSTCA.crl”
* “openssl crl –text –noout –in intermediate/crl/CSTCA.crl”

1. Understand how we can use Online Certificate Status Protocol (OCSP) to determine revocation status of a certificate over the network.

* “openssl genrsa –aes256 –out intermediate/private/ocsp.key 4096”
* “openssl req –config intermediate/openssl.cnf –new –sha256 key intermediate/private/ocsp.key –out intermediate/csr/ocsp.csr”
* “openssl ca –config intermediate/openssl.cnf –extensions ocsp –days 375 –notext –md sha265 –in intermediate/csr/ocsp.csr –out intermediate/certs/ocsp.cert”
* “openssl ocsp –port 127.0.0.1.2560 –text –sha256 –index intermediate/index.txt –CA intermediate/certs/CA-Chain.cert –rkey intermediate/private/ocsp.key –rsinger intermediate/certs/ocsp.cert –nrequest 1”
* “openssl ocsp –CAfile intermediate/certs/CA-Chain.cert –url <http://127.0.0.1:2560> –issuer intermediate/certs/CSTCA.cert –cert intermediate/certs/bob.cert”
* “opens ocsp –CAfile intermediate/certs/CA-Chain.cert –url <http://127.0.0.1.2560> –issuer intermediate/certs/CSTCA.cert –cert intermediate/cert/darth.cert”