Vejledning til brug af klient-certifikater i den grønlandske E-boks løsning

Den grønlandske E-boks løsning gør brug af gensidig SSL validering til at godkende de klienter der skal anvende proxyen. Det vil sige at serveren for løsningen kører HTTPS og præsenterer et server-side certifikat som klienten skal verificere og at klienter skal præsentere et klient-certifikat over for serveren, som serveren så validere inden der gives adgang.
Denne vejledning giver et kort indblik i hvordan man får anskaffet og registreret et klient-certifikat og hvordan man anvender det i kald til løsningen

Forudsætninger

Klient-certifikat

For at kunne gøre brug af services i den grønlandske e-boks proxy skal der bruges et klient-certifikat når der sendes forespørgsler til proxyen.
For test-miljøet får man et sådant certifikat og dets tilhørende nøgle sendt fra Magenta når der bliver fortaget en klient registrering. Dette certifikat vil allerede være godkendt af test-instancen af proxyen og kan anvendes umiddelbart.

I produktionsmiljøet vil man skulle anskaffe et FOCES certifikat fra nemid[1]. Bemærk at det kun er nødvendigt at sende selve certifikatet og ikke den tilhørende nøgle. Certifikater leveres normalt i en .pfx-fil der indeholder både certifikat og nøgle. Nøglen bør holdes hemmelig og kun findes hos klienten, så det vil være nødvendigt at trække certifikatet ud af .pfx-filen og kun sende certifikatet.
Klient-certifikatet vil kunne anvendes når Magenta melder tilbage om at det er blevet registreret i produktions-proxyen.

Accept af server-side HTTPS certifikat

Hvis man benytter den interne adgang[2] (både for test og for produktion) benytter proxy serveren et HTTPS certifikat udstedt af KNNO root CA'en. Det vil sige at klient-software der laver kald til serveren skal stole på KNNO root CA'en for ikke at få HTTPS-valideringsfejl. Det kan derfor være nødvendigt at konfigurere ens https klient og eller operativsystem til at stole på certifikater udstedt af KNNO root CA`en.

---

1 Se afsnittet "Adgang til produktion" i dokumentet: Teknisk vejledning til anvendelse af Grønlandsk Proxy til e-Boks kommunikation

2 Se afsnittene "Adgang til produktion" og "Adgang til test" i dokumentet: Teknisk vejledning til anvendelse af Grønlandsk Proxy til e-Boks kommunikation

Vejledning

For at kommunikere med E-boks proxyen er det nødvendigt man har en https klient/bibliotek der understøtter gensidig ssl validering.Der findes ofte flere forskellige biblioteker til forskellige programmeringssprog, her er nogle eksempler:

- **.Net**: System.Net.Http.HttpClient
- **Java**: java.net.HttpURLConnection, java 9: jdk.incubator.http.HttpClient
- **Python**: requests, urllib2
- **PHP**: Client URL Library
- **Ruby**: Net::HTTP, rest-client, httparty

Opsætning af validering af HTTPS-certifikat fra KNNO root CA'en

Som nævnt tidligere kan det være nødvendigt at klienten stoler på KNNO root CA'en. Den mest almindelige måde at gøre dette på er ved at installere ROOT CA'en som en af de CA'er operativ-systemet eller klienten stoler på. Følgende tutorial viser hvordan man gør dette på forskellige operativ-systemer og platforme:

https://www.bounca.org/tutorials/install_root_certificate.html

Bemærk at Windows-maskiner i KNNO-miljøet i Grønland allerede har certifikatet installeret og automatisk vil stole på certifikatet.
En kopi af root CA certifikatet i PEM format kan findes i Bilag 1 i dette dokument.

Anvendelse af klient-certifikat

De forskellige biblioteker til forskellige sprog har forskellige måder at håndtere brugen af et klient-certifikat til forespørgsler, men typisk vil skulle gøre noget lignende:

- Loade klient-certifikat og dets nøgle ind som objekter i ens kode
    - Java og .Net vil typisk kræve at certifikatet og dets nøgle først er blevet importeret til en certifikat-store og at de loades der fra, frem for at loade certifikatet ind fra en fil på disk
- Tilføje certifikat og nøgle til transport-laget i ens http-klient
- Eventuelt: Konfigurer klienten til at acceptere certifikater underskrevet af KNNO CA'en
- Foretage forespørgsler som normalt gennem den konfigurerede klient

Her er et eksempel på hvordan man gør det via requests biblioteket i Python:

```
import requests

# Create a sesssion
session = requests.Session()
# Enable usage of client certificate for the session
session.cert = ("./myclientcertificate.cert", "./myclientcertificate.key")
# Set the client to explicitly trust the KNNO root CA
session.verify = "./KNNO_root_CA.pem"

# Send request to bogus path, should return 404 Not Found
res = session.get("https://noeboxproxy01.dmz70.local/rest/some/path")
print(res)
```

Hvis man under udviklingen får fejlbeskeder relateret til brug af certifikater skal man være opmærksom på om der er tale om en fejl i valideringen af server-side certifikatet (HTTPS forbindelsen til serveren), eller om fejlen er relateret til brugen af klient-certifikat. Et forkert eller ikke-godkendt klient-certifikat vil resultere i en 403 Forbidden HTTP fejlkode fra webserveren, mens fejl i valideringen af serverens HTTPS certifikat vil give beskeder om dette inden forbindelsen til serveren er færdigoprettet.

Bilag 1: KNNO root CA certifikat

```
-----BEGIN CERTIFICATE-----
MIIDBzCCAe+gAwIBAgIQVKnANbw5QKFOe26SGHVYJTANBgkqhkiG9w0BAQsFADAW
MRQwEgYDVQQDEwtLbm5vLVJvb3RDQTAeFw0xOTA5MjQxODM5MjNaFw0zOTA5MjQx
ODQ5MDVaMBYxFDASBgNVBAMTC0tubm8tUm9vdENBMIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEA3WW8YvhACqWmL5sisC0MBTZ+s41dxpvkl3ldQDqLcbWZ
/1tofEUyxfWYWXaywqMWr1jEauhfyCCZkuP6XpdBB5XTllq7wxUf6Kobd1wWbjlC
HreHnJ4cJSUo9+FTq24v3DNwuPdkZE3FBkjln0ZcYBqkcEWkMa85x0nTsSrzXh9H
ou19GcFKWmazsdqRLMBrqtSk6Zhyvof0XWiplHIb2Imc91NNGAt/q4112n+KtXs2
W2fimKa/zpy17XmLTodLebX+FaFJNwTze+0amb0yP9i9Q+iJwVqDFOapvWuw1jTT
cZNFadg6M666JyDLuFGXy47Ze235vZncEoUmPRSMtQIDAQABo1EwTzALBgNVHQ8E
BAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUfNuobz2SrODs/BMZKidi
5++dAlQwEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQELBQADggEBAHMP3V7c
y7WGzoJrjbvFJJC0Sxlh/q91tGT+jG2P0rPrfcaSxJqL36TKimcA9F6vKz3O96SA
hqF7QXwARQMpMwEy8EfwV5WMYZpSRHInNzgZTnN2v+5ZtEo2Ox4hFzVFWSzE1YCN
LpZt/6WGclq2mI2EiTlyvptTPFyoYMho/yim6u4ECM/h8i8WBsOyfwI3cPehMTTp
EV2NL0gDcNYVWR/WB5ZTOXfpB634LwVc5V1P6dtSXEDw+4Sxg3qZJHCepoB5HxaW
6MtddbXsnrCOURfZK2WmfA22p4vtKDwmt5pgagx9ZJ4ynUDUcUeb3BEAQlz7wgKk
ZmhOC2MJJyCJYlQ=
-----END CERTIFICATE-----
```