FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Voorbereiding Labo Certificaten

Bert De Saffel

Master in de Industriële Wetenschappen: Informatica Academiejaar 2018–2019

UNIVERSITEIT
GENT

Inhoudsopgave

Hoofdstuk 1

Vragen

- **Wat is de functie van een certificaat?** Een certificaat bevat informatie over de eigenaar van dit certificaat zoals e-mailadres, naam, geldigheidsdatum. Ook bevat het een publieke sleutel en een hashwaarde om na te gaan of dit certificaat niet gewijzigd werd. Het doel van zo een certificaat is om na te gaan of dat een website waarop iemand zich bevindt, wel degelijk de site is dat hij verwacht dat het is. Een certificaat wordt getekend door een erkende partij (Certificate Authority). De CA heeft een lijst van alle getekende en verworpen certificaten. Certificaten die nog niet getekend zijn, worden als onveilig beschouwd aangezien die nog aangepast kunnen worden. Een getekend certificaat kan niet aangepast worden.
- **Welke stappen moet je volgen om een certificaat aan te vragen, indien het subject een server-programma is?** De server moet een **Certificate Sign Request (CSR)** genereren dat ondertekend moet worden door een CA. Een CSR bevat onder andere de publieke sleutel en identiteitsinformatie (zoals domainnaam). Vooraleer een server zo een CSR kan aanmaken, moet hij eerst een sleutelpaar genereren. De server moet zijn CSR ondertekenen met zijn geheime sleutel.

Hoofdstuk 2

OpenSSL

OpenSSL is een library dat onder andere gebruikt wordt door openSSH.

2.1 Opties

Elk openssl commando wordt gevolgd door een sleutelwoord. Een sleutelwoord kan dan gevolgd worden door eventuele opties, specifiek voor dat sleutelwoord. Een overzicht van de belangrijkste sleutelwoorden zijn:

- **req** Dit sleutelwoord zal een aanvraag sturen om een certificaat te ondertekenen. Eventuele opties zijn:

  -x509 Deze optie zal een zelf getekend certificaat (self signed certificate) geven in plaats van het te vragen aan een erkende partij. Dit wordt typisch gebruikt voor testcertificaten.

  -nodes Bij deze optie zal de geheime sleutel niet geëncrypteerd worden.

  -days n In combinatie met de **-x509** optie, zal deze optie het certificaat voor **n** dagen geldig houden. De defaultwaarde voor **n** is 30.

  -newkey arg Deze optie zal een nieuwe geheime sleutel genereren alsook een nieuwe certificaataanvraag starten. Het argument kan drie vormen aannemen:

  * **rsa:nbits** Dit genereert een **RSA** sleutel van grootte **nbits**. Indien nbits weggelaten wordt, zal de defaultwaarde genomen worden van het configuratiebestand.
  * **param:file** Dit zal een sleutel genereren met een parameter- of certificaatbestand. Het algoritme wordt bepaald door de parameters in dit bestand.
  * **[dsa|ec|gost2001]:filename** zal respectievelijk een **DSA**, **EC** of een **GOST** sleutel genereren, gebruik makend van de parameters in het bestand **filename**.

  -keyout filename Deze optie specifieert het bestand waar de geheime sleutel naartoe moet geschreven worden.

  -out filename Deze optie specifieert naarwaar de output van het commando moet geschreven worden.

  -verify Verificatie van de handtekening van de aanvraag.

  -text Zal de aanvraag in textformaat uitprinten

Voorbeelden van het gebruik van **req**:

- **Controle en verificatie certificaat aanvraag**

```
openssl req -in req.pem -text -verify -noout
```

- **Geheime sleutel aanmaken en daarna deze gebruiken voor een nieuwe certificaataanvraag**

```
openssl req -newkey rsa:1024 -keyout key.pem -out req.pem
```

- **Genereren van een self signed root certificaat**

```
openssl req -x509 -newkey rsa:1024 -keyout key.pem -out req.pem
```

- **genrsa** Dit sleutelwoord kan als vervanging van **req -newkey rsa:nbits** dienen. Voorbeelden van het gebruik van **genrsa**:

  - **Geheime sleutel aanmaken en daarna deze gebruiken voor een nieuwe certificaataanvraag**

```
openssl genrsa -out key.pem 1024
openssl req -new -key key.pem -out req.pem
```