Certificados digitais

Esta prática de laboratório visa explorar o uso de certificados digitais X.509 no âmbito da Web.

Explorando certificados

Acesse os sites Web da lista abaixo e analise os certificados que eles oferecem ao navegador:

- https://www.itau.com.br
- https://sites.google.com
- https://listas.inf.ufrgs.br

Para cada site acessado, responda às seguintes questões:

- quem emitiu o certificado?
- qual o período de validade do certificado?
- qual a finalidade do certificado (conforme informado no campo de extensão `Key Usage`, se estiver presente)?
- qual a cadeia de certificação estabelecida?

Explorando cadeias de certificação

O utilitário OpenSSL oferece várias funcionalidades para trabalhar com certificados SSL/TLS. Por exemplo, o comando abaixo permite visualizar a cadeia de certificação de um determinado serviço de rede:

```
openssl s_client -showcerts -connect www.server.com:port_number
```

Usando esse programa, analise a cadeia de certificação dos sites indicados no exercício anterior (sites HTTPS costumam usar a porta 443). Existem informações que não haviam sido encontradas anteriormente?

Criando uma CA e assinando certificados

Este roteiro (adaptado deste site) compreende três “atores”: uma CA, um servidor Web e um cliente (navegador). Eles executam as seguintes ações, em sequência:

1. CA: criação dos arquivos da Autoridade Certificadora fictícia
2. Servidor: criação de um par de chaves para o serviço Web e da requisição de certificado
3. CA: assinatura do certificado do serviço Web
4. Servidor: instalação do certificado
5. Cliente: navegação e acesso ao serviço

1) Criar uma autoridade certificadora

Criar a estrutura de diretórios e arquivos usados por nossa CA fictícia (de acordo com os valores *default* definidos em `/usr/lib/ssl/openssl.cnf`):

```
mkdir demoCA
mkdir demoCA/newcerts
```

```
touch demoCA/index.txt
echo "01" > demoCA/serial
```

Gerar a **chave privada** e o **certificado digital** de nossa CA "raiz" (auto assinado), com validade (por exemplo) para 10 anos:

```
openssl req -new -x509 -keyout ca.key -out ca.crt -days 3650
```

O comando acima irá solicitar uma série de informações que serão incluídas no certificado e uma senha para a proteção da chave privada:

```
Enter PEM pass phrase: [ digite uma senha para proteger a chave privada ]
Verifying - Enter PEM pass phrase: [ repita a senha ]
-----
Country Name (2 letter code) [AU]: BR
State or Province Name (full name) [Some-State]: Parana
Locality Name (eg, city) []: Curitiba
Organization Name (eg, company) [Internet Widgits Pty Ltd]: UFPR
Organizational Unit Name (eg, section) []: Certification Authority Dept
Common Name (e.g. server FQDN or YOUR name) []: ca.ufpr.br
Email Address []:meu.email@ufpr.br
```

Após a execução serão gerados dois arquivos:

- `ca.key` contém a chave privada de nossa CA, protegida por senha.
- `ca.crt` contém o respectivo certificado de chave pública auto-assinado.

O certificado é gerado em no formato PEM (*Privacy Enhanced Mail*). Pode-se visualizar o conteúdo do **certificado** recém-gerado (arquivo `ca.crt`) em um formato textual usando:

```
openssl x509 -inform pem -in ca.crt -text
```

Da mesma forma, pode-se verificar o conteúdo da **chave privada** recém-gerada (arquivo `ca.key`) usando:

```
openssl rsa -inform pem -in ca.key -text
```

A seguir, devemos mover o certificado e a chave privada da CA para o local adequado (diretório da CA):

```
mv ca.crt ca.key demoCA/
```

2) Criação das chaves do serviço Web

Inicialmente, vamos criar um diretório para o servidor:

```
mkdir server
```

Em seguida, gerar um **par de chaves** pública/privada para o servidor; uma senha será solicitada para proteger a chave privada dos servidor:

```
openssl genrsa -aes128 -out server/server.key 2048
```

O conteúdo do arquivo `server.key` pode ser inspecionado através deste comando:

```
openssl rsa -in server/server.key -text
```

A partir do par de chaves do servidor, deve ser gerada uma **requisição de assinatura** de certificado (*Certificate Signing Request* - CSR), a ser enviada à CA:

```
openssl req -new -key server/server.key -out server.csr
```

Uma série de informações sobre o certificado será solicitada. Neste caso, use `localhost` para o *Common Name* do servidor Web; em um caso real, deveria ser usado o FQDN do servidor (por exemplo, `www.servidor.com.br`):

```
Enter pass phrase for server/server.key: [ digite a senha da chave privada do
servidor ]
-----
Country Name (2 letter code) [AU]: BR
State or Province Name (full name) [Some-State]: Parana
Locality Name (eg, city) []: Curitiba
Organization Name (eg, company) [Internet Widgits Pty Ltd]: UFPR
Organizational Unit Name (eg, section) []: DINF
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:maziero@inf.ufpr.br
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: [deixe em branco]
An optional company name []: [deixe em branco]
```

O arquivo `server.csr`, contendo a requisição em formato PEM, deve ser enviado à CA para ser assinado digitalmente. Esse envio pode ser feito por e-mail ou outro meio, mesmo que inseguro.

3) Gerar o certificado do servidor Web

Quando a CA recebe uma requisição de assinatura de certificado (arquivo `server.csr`), ela deve primeiro criar um arquivo `server.ext` contendo informações adicionais (*certificate extensions*) sobre o certificado a ser criado, com o seguinte conteúdo:

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
subjectAltName = @alt_names
[alt_names]
DNS.1 = localhost
```

A seguir, a CA gera um **certificado assinado** usando sua chave privada (`ca.key`) e seu próprio certificado (`ca.crt`):

```
openssl ca -policy policy_anything -in server.csr -out server.crt -cert
demoCA/ca.crt -keyfile demoCA/ca.key -extfile server.ext
```

O certificado do servidor assinado pela CA (`server.crt`) pode ser visualizado através do seguinte comando:

```
openssl x509 -inform pem -in server.crt -text
```

A CA então envia o certificado assinado por ela (`server.crt`) de volta para o requerente. Os arquivos `server.csr` `server.ext` podem ser descartados:

```
rm server.csr server.ext
```

4) Instalando o certificado assinado no servidor Web

Os arquivos `server.crt` (certificado assinado pela CA) e `server.key` (chave privada do servidor) podem então ser instalados no servidor Web, de acordo com a configuração de cada servidor (Apache, etc).

```
mv server.crt server/
cd server
```

Neste laboratório, podemos simular um servidor Web usando o próprio comando `openssl`, que atende HTTPS na porta 4433, com o certificado que acabamos de gerar. Para isso, primeiro devemos concatenar o certificado e a chave privada em um só arquivo:

```
cat server.key server.crt > server.pem
```

Agora podemos lançar o servidor Web seguro:

```
openssl s_server -cert server.pem -www
```

5) Acessar o servidor Web

Acesse o servidor Web seguro (https://localhost:4433) no navegador. O que aconteceu?

Para que o navegador aceite o certificado do site Web seguro, ele deve ter o certificado da entidade certificadora (CA) armazenado em sua base de certificados confiáveis.

- Inclua o certificado de nossa CA fictícia (arquivo `demoCA/ca.crt`) no repositório de certificados do navegador.

Agora, acesse novamente o servidor Web seguro. O que mudou?