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 e deste site) compreende a criação dos arquivos básicos que definem uma Autoridade Certificadora (CA), a criação de um par de chaves para um servidor Web e a geração do certificado assinado correspondente.

Criando uma autoridade certificadora

Criar a estrutura de diretórios e arquivos usados por nossa CA (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 arquivo `ca.key` contém a chave privada de nossa CA e o arquivo `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 em um formato textual usando:

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

No servidor Web que deseja um certificado

Inicialmente, gerar um par de chaves pública/privada para o servidor:

```
openssl genrsa -aes128 -out server.key 1024
```

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

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

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

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

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`).

O arquivo `server.csr`, 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.

A CA vai assinar um certificado

Quando a CA receber uma requisição de assinatura de certificado, ela 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
```

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.

Lançando o 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. O arquivo `server.csr` pode ser descartado, pois não tem mais utilidade.

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:

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
```

Enfim, no navegador

- Acesse o servidor Web seguro (https://localhost:4433). 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. Por isso, inclua o certificado de nossa CA (arquivo `demoCA/ca.crt`) no repositório de certificados do navegador.
- Acesse novamente o servidor Web seguro. O que mudou?