Licenciatura em Eng.ª Informática
Complementos de Redes - 3º Ano - 2º Semestre

Trabalho Nº 2 - Gestão de Certificados e SSL/TLS

1. Introdução

A segurança baseada em Criptografia de Chave Pública assenta na geração e distribuição de Certificados Digitais, através dos quais uma CA - Certification Authority garante a identidade dos utilizadores. O conjunto de regras, entidades, protocolos e artefactos necessários para estabelecer condições de segurança baseadas neste tipo de tecnologia designa-se por PKI - *Public Key Infrastructure*.

No decorrer deste trabalho iremos utilizar as funcionalidades do OpenSSL, uma biblioteca e utilitários de segurança distribuídos livremente com sistemas Linux, para criar uma PKI entre os alunos do laboratório e uma CA criada pelo professor. Seguidamente será utilizada a API OpenSSL.

2. Trabalho a efectuar

2.1 Geração de um certificado

No decorrer deste trabalho, cada aluno irá gerar um certificado digital individual associado a uma chave de encriptação privada, que será depois validado através da aposição de uma assinatura digital por uma CA.

Para tal deverá proceder do seguinte modo:

1. Criar uma pasta específica para colocar os certificados dentro da sua área de trabalho:

```
mkdir trabalho2
mkdir trabalho2/certs
```

2. Criar uma chave privada RSA de encriptação com um módulo de 2048 bits utilizando o comando openssl, guardada num ficheiro que deverá ter o nome **key.pem**:

```
cd trabalho2/certs
openssl genrsa -out key.pem 2048
```

3. Com base nessa chave, irá gerar um *Certificate Signing Request* (CSR) utilizando também o comando *openssl*. Um CSR só será aceite como certificado depois de ter sido assinado por uma Certification Authority (CA).

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

No decorrer da execução desse programa, será estabelecido um diálogo na consola em que o comando irá pedir a identificação do **subject**, neste caso o aluno. Deverá indicar os seus dados, incluindo uma password para protecção (que convém não esquecer) seguindo o exemplo a seguir apresentado:

```
linux$ openssl req -new -key key.pem -out csr.pem
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:PT
State or Province Name (full name) [Some-State]:Estremadura
Locality Name (eg, city) []:Lisboa
Organization Name (eg, company) [IBM]:Universidade Lusofona
Organizational Unit Name (eg, section) []:Licenciatura Eng.
Informatica
Common Name (eg, YOUR name) []:Jose Rogado
Email Address []:jrogado@ulusofona.pt

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:*****
An optional company name []:ULHT
linux$
```

4. A seguir, o ficheiro **csr.pem** deverá ser enviado ao professor que irá proceder à sua assinatura com a chave privada da Certification Authority. Só depois o seu pedido de certificado será transformado num certificado válido.

5. Ao receber de volta o certificado **crt.pem**, deverá colocá-lo na mesma pasta onde está a sua chave privada. Poderá ver as suas características com o comando:

   ```
   openssl x509 -issuer -subject -dates -in crt.pem
   ```

6. Seguidamente deverá realizar o download do ficheiro **TrustStore.pem** que contém os certificados das Autoridades de Certificação mais comuns, incluindo a da CA do Laboratório criada para este trabalho. O ficheiro está disponível em:

   http://netlab.ulusofona.pt/cr/praticas

   Guarde esse ficheiro na pasta **trabalho2/certs** e abra-o com um editor de texto para identificar os certificados das CAs que contém e as suas datas de validade.

7. Seguidamente iremos utilizar o certificado para proceder a uma conecção segura com alguns dos seus colegas.

2.2 Utilização do Certificado

A utilização do certificado pessoal irá ser feita de duas formas:

- Utilizando o comando *openssl* para estabelecer uma conecção encriptada entre um cliente e um servidor.
- Através de um programa (C ou Java) a desenvolver pelo aluno que irá realizar o papel do cliente e interagir com o servidor openssl.

Em qualquer dos casos, o certificado criado previamente irá ser utilizado pelo servidor, sendo o cliente responsável pela sua validação, utilizando a chave pública da CA contida no respectivo certificado (recebido com o ficheiro *TrustStore.pem*).

2.2.1 Utilização do comando *openssl*

Para realizar uma conecção encriptada entre dois intervenientes, deverá proceder da seguinte forma:

1. Abra uma consola para o servidor e coloque-se na pasta onde guardou os certificados criados no ponto anterior. Digite o comando seguinte:

   ```
   openssl s_server –cert crt.pem –key key.pem
   ```

2. Abra uma segunda consola para o cliente, coloque-se na mesma pasta e digite o comando seguinte:

   ```
   openssl s_client –CAfile TrustStore.pem
   ```

3. Interprete as mensagens recebidas pelo cliente e servidor. Pode verificar que tudo quanto escrever no terminal do cliente é recebido pelo servidor e vice-versa, através de uma mensagem que circula cifrada pela rede.

4. Realize uma conecção com a estação de trabalho de um colega seu, utilizando desta vez a opção seguinte, em que *host* é o endereço IP respectivo e :*port* o 4433 por defeito:

   ```
   openssl s_client –connect host:port –CAfile TrustStore.pem
   ```

5. Identifique as mensagens trocadas entre o cliente e o servidor com o programa Ethereal e verifique que os dados enviados estão efectivamente encriptados. Capture essas mensagens e inclua o respectivo SnapShot no relatório.

2.2.2 Programação do Cliente

Implemente um programa em C ou Java que realize a mesma funcionalidade que o cliente previamente simulado com o **openssl**.

Esse programa deverá:

1. Utilizar a biblioteca OpenSSL e a respectiva API

2. Criar um contexto SSL

3. Verificar a existência do ficheiro TrustStore.pem e carregar os certificados das CAs que contém

4. Abrir uma conecção SSL para o servidor indicado na linha de comandos através da sintaxe **host:port**

5. Verificar se o certificado recebido do servidor nessa conecção pode ser autenticado através de um dos certificados contido no TrustStore

6. Ler uma linha do *standard input* e enviá-la para o servidor *openssl*, que deverá recebê-la correctamente.

Para a realização deste programa em C, é aconselhável seguir o tutorial "*Secure Programming with the Open SSL API*", disponível em:

http://www-128.ibm.com/developerworks/linux/library/l-openssl.html

Para a realização do mesmo programa em Java, aconselha-se a leitura da documentação disponível em:

http://java.sun.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html

Esta documentação contém uma excelente descrição do protocolo SSL/TLS sendo aconselhada a sua leitura mesmo para quem realizar o programa em C.

3. Prazo de Entrega do Trabalho

O prazo para a entrega dos trabalhos é de duas semanas a contar da aula em que o enunciado foi apresentado[1]. Não serão aceites trabalhos fora do prazo. A entrega deverá ser feita por e-mail num ficheiro zip (ou rar) contendo um relatório (**obrigatório**) em PDF e as listagens dos programas realizados, obedecendo OBRIGATORIAMENTE ao seguinte formato:

`a123456-trabalho-N.zip`

4. Referências

"The Open Source toolkit for SSL/TLS":

www.openssl.org

"Secure Programming with the Open SSL API":

http://www-128.ibm.com/developerworks/linux/library/l-openssl.html

"JSSE Reference Guide for Java SE 6":

http://java.sun.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html

[1] No caso particular deste trabalho, a entrega será na primeira aula prática depois das férias da Páscoa