PRÁCTICA 7

Supuesto práctico:

Alice está ejecutando el servidor Web Apache y tiene una carpeta privada con imágenes de un gato muy bonito. Alice quiere conceder a su amigo Bob, el acceso a esta carpeta. Para esto, es necesario un **certificado digital**.

Cuando ALice desea conceder acceso a Bob, es fundamental asegurarse de que Bob es quien dice ser. Un certificado digital cumple con dicha función, proporcionando un medio seguro y verificable de autenticación, tanto para el servidor como para el cliente.

Para Alice: un certificado digital (el certificado (autocertificado)) firmado por una Autoridad de Certificación asegura que el servidor web de Alice es legítimo y que no es fraudulento o interceptador (lo que se conoce como un ataque de Man In the Middle).

Para Bob: el certificado digital de Bob sirve para que Alice sepa que el usuario que está intentando acceder a las imágenes es realmente Bob, y no alguien que esté intentando suplantar su identidad.

Elementos principales de un certificado digital: Un certificado digital es un documento electrónico que utiliza tecnologías criptográficas para autenticar la identidad de una entidad (como un sitio web, persona, organización o dispositivo) y garantizar la seguridad de las comunicaciones digitales. Es emitido y firmado por una Autoridad de Certificación (CA).

- Información de la entidad (Sujeto):
    - Nombre de la organización o persona.
    - Nombre del dominio o dirección web (en el caso de certificados SSL/TLS).
    - Información de contacto, como correo electrónico.
- Clave pública:
    - Asociada a la entidad y utilizada en criptografía de clave pública para cifrar datos o verificar firmas.
- Emisor (Issuer):
    - La CA que emitió el certificado. Esta es una entidad confiable que verifica la identidad del sujeto antes de emitir el certificado.
- Periodo de validez:
    - Fecha de inicio y fecha de expiración del certificado.
- Número de serie:
    - Identificador único del certificado.

- Algoritmo de firma:
    - Especifica cómo la CA firma el certificado para garantizar su autenticidad.
- Extensiones opcionales:
    - Pueden incluir políticas de uso, restricciones, o información adicional sobre el propósito del certificado.

BOB

Bob comienza solicitando un certificado digital. Para ello, Bob genera su clave privada y crea una solicitud de firma de certificado (CSR):

```
cd bob

openssl genrsa -out bob@example.com.key.pem 2048
```

A continuación, procede a crear la **solicitud de firma de certificado (CSR)**:

```
openssl req -new -key bob@example.com.key.pem -out bob@example.com.csr.pem
```

Con esta sentencia creamos una nueva (-new) solicitud de firma de certificado (CSR), pero, **¿qué es una CSR?**:

> Una Solicitud de Firma de Certificado (CSR), es un archivo que contiene los siguientes elementos:
>
> - Datos del solicitante: Toda la información identificativa que se quiere incluir en el certificado, que tendremos que meter a mano al ejecutar la orden.
> - La clave pública de Bob, que se obtiene a partir de la clave privada recién generada.
> - La firma digital del contenido de la CSR: es decir, se cogen todos los datos (información del solicitante + clave pública), se le hace un hash y se cifra con la clave privada de Bob. Así, cuando la autoridad de certificación intermedia reciba la solicitud, como tiene la clave pública, y los datos, puede verificar la auteticidad del emisor y la integridad del mensaje.

A continuación mostramos una imagen en la que podemos ver la **estructura de una CSR**:

bob@example.com
Solicitud de certificador
Identidad: bob@example.com

Detalles

Nombre del asunto

C (País):	MA
ST (Estado):	Casablanca
L (Localidad):	Casablanca
O (Organización):	Bob Ltd
CN (Nombre común):	bob@example.com

Solicitud de certificador

Tipo:	PKCS#10
Versión:	1

Información de la clave pública

Clave del algoritmo:	RSA
Parámetros de la clave:	05 00
Tamaño de la clave:	2048
Huella de la clave SHA1:	7C 58 57 92 59 16 A0 58 76 21 F0 82 74 9C E4 B6 31 84 F4 58
Clave pública:	30 82 01 0A 02 82 01 01 00 F2 96 29 49 6E 82 20 DF 9C 52 FE F6 46 2B E5 9D 9A 86 BA 67 37 14 3C BB 17 1E D5 20 D9 06 07 23 3B 2C 72 51 A6 5C C0 39 0E B0 E9 C8 C6 7A 98 7F 04 FE FC A2 39 45 28 67 71 EA 17 26 E4 B8 F3 F5 33 74 BC BF A2 0E 5D 49 2B 76 1C 12 68 AB 5C 75 01 45 64 BF 9F 7B 91 EF 59 9F FF AD 26 1C D9 B9 61 A9 F7 1D AD 6A F4 7D DF 93 6B 75 F4 07 50 5D 72 E9 A9 4E 88 A6 86 85 B2 92 70 B5 36 45 B8 53 22 C5 67 59 26 8A 27 FC 0B AF D7 B9 AC C0 B5 15 B0 8B CA D5 47 0B 17 64 22 9C CC E1 BD 7F E1 61 D4 30 94 70 32 8C E6 1A 14 8B 0C 62 03 53 B7 40 DD 78 77 8A 43 DF 6E 75 F1 1F 75 9B 81 96 6F 6B DA 40 79 B3 5E 70 1A 6E F1 A4 17 2C 36 10 4F DF DE A2 8D A7 A5 61 3D EF 2B 38 29 36 90 DB 2C EF 67 57 A8 FA DC AE 99 08 B3 32 FA DC 1F 85 E7 6A ED B8 D9 FA AE E5 88 40 42 FF B7 63 6F 23 2C 19 02 03 01 00 01

Atributo

Tipo:	1.2.840.113549.1.9.2
Valor:	0C 07 42 6F 62 20 4C 74 64

Atributo

Tipo:	
Valor:	0C 07 73 75 43 6C 61 76 65

Firma

Algoritmo de firma:	1.2.840.113549.1.1.11
Parámetros de la firma:	05 00
Firma:	B6 30 5E 1E BA D7 E4 41 8C A9 D9 0C 5C 90 1A ED F8 D8 31 0E D5 C8 40 12 AB 18 47 30 9E 68 3A E2 22 23 33 00 C1 B8 D3 44 B3 85 8C 59 5A 88 C8 92 0A 4B 07 98 15 36 3C E2 C5 B7 69 8C 18 4E DB 89 D3 DE 17 A9 BF 0C 08 80 78 2D 4B 20 19 33 C5 C2 83 D2 80 0C 52 77 6D 70 B5 3E C3 60 7D FD 11 62 49 29 80 AB CD 77 1D 0A 98 8C 68 4E AF E4 8D 73 EE A6 5C ED 27 C8 93 81 CF 9C 6E 67 1D 51 F9 7C 78 65 EA 64 BB A0 58 71 50 69 F2 D0 5E AA DD 21 4D 85 0D 7A 05 2A 6A B6 76 36 7C 01 C1 1A 6D 5C 4E 60 EC 7C 3F BB 38 2D AC 05 2D B5 91 72 E4 AA 01 47 4A D6 4D 96 67 92 B5 87 FD 86 6D E9 2F A3 39 9F 00 8F 74 A5 F7 55 BE 3D DB 65 53 A5 2F 37 F3 54 13 3D 38 4A 81 45

Observamos que están presentes los datos del solicitante, la clave pública y la firma digital que se hizo con la clave privada de Bob.

Una vez que ya se ha creado la CSR, Bob envía su solicitud a Alice.

ALICE

Recepción de CSR y firma (emisión del certificado)

Cuando Alice recibe la CSR, lo firma:

```
openssl ca -config intermediate/openssl.cnf -extensions usr_cert -notext -
md sha256 -in intermediate/csr/bob@example.com.csr.pem -out
intermediate/certs/bob@example.com.cert.pem -subj "/CN=bob@example.com"
```

Indicamos a OpenSSL, desde root/ca con función **ca**, que actuaremos como una Autoridad de Certificación (CA) y que emitirá un certificado a partir de una CSR (firmará una CSR).

Con **-config intermediate/openssl.cnf**, indicamos que se debe utilizar el archivo de configuración intermediate/openssl.cnf. Este archivo contiene parámetros y configuraciones necesarias para la firma del certificado, como las políticas, extensiones y configuraciones de directorios de la CA.

La opción **-extensions usr_cert** especifica qué extensiones de certificado deben ser aplicadas al certificado. Las extensiones son atributos adicionales que se incluyen en el certificado. En este caso, se está utilizando la extensión **usr_cert**, que normalmente está configurada en el archivo de configuración para certificados de usuario estándar.

La opción **-notext** se refiere a que el comando no incluirá una representación legible por humanos del certificado en el archivo de salida.

Con **-md sha256**, se especifica el algoritmo de hashing a utilizar para firmar el certificado que le vamos a otorgar a Bob.

Con **-in intermediate/csr/bob@example.com.csr.pem**, se especifica el archivo de la CSR que será procesado, el cual tiene la clave pública de Bob, los datos identificativos de Bob y la firma. La autenticidad de Bob y la integridad del mensaje son verificados, para lo cual, se usa la clave pública que está almacenada en este archivo (la clave pública incluida en la CSR tiene una clave privada correspondiente que solo Bob debería poseer, y el mensaje no se ha podido modificar por el camino).

Con **-out intermediate/certs/bob@example.com.cert.pem** especificamos el archivo de salida donde se guardará el certificado firmado (firmado por la autoridad de certificación intermedia, con su clave privada, intermediate.key.pem).

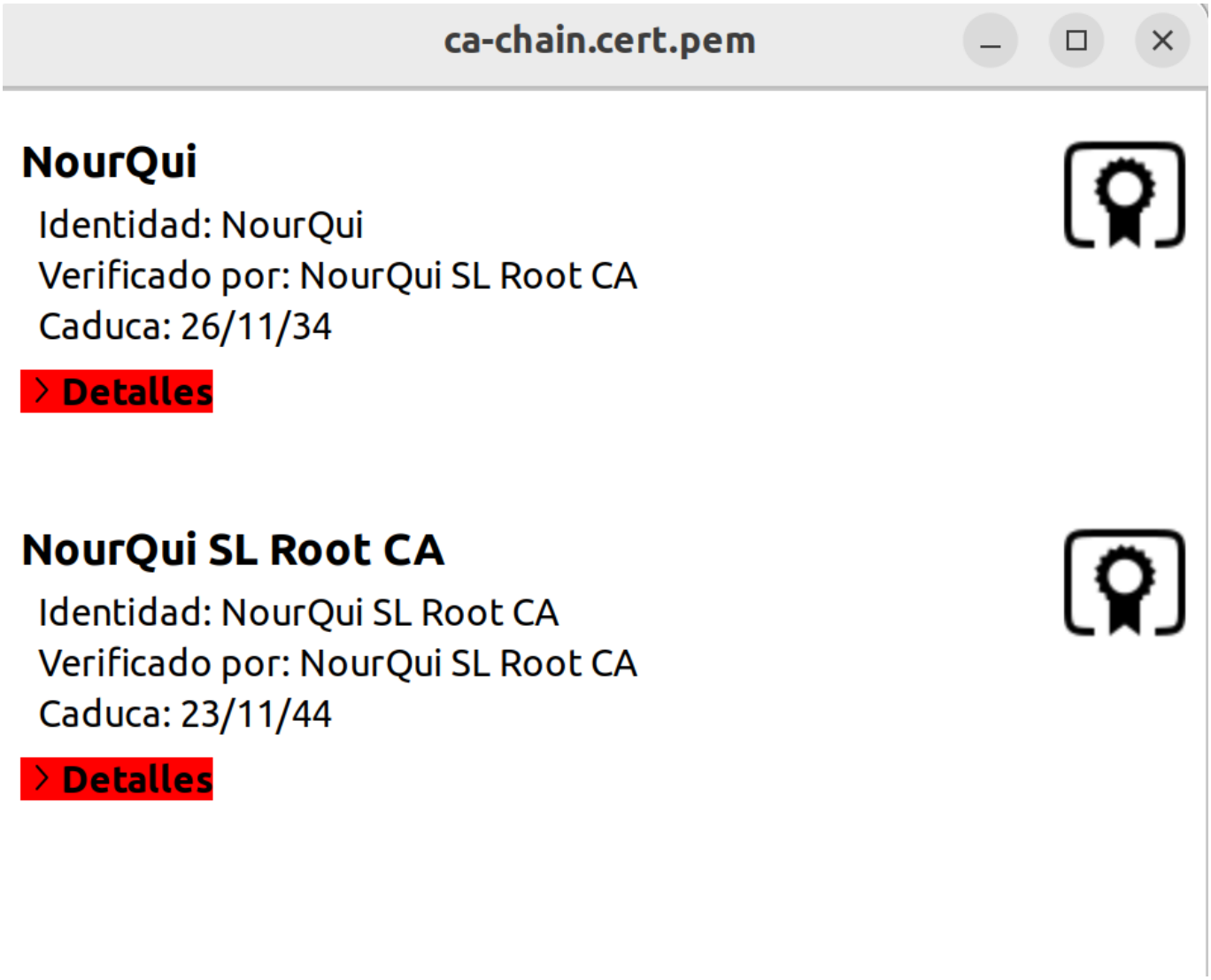
Finalmente, con la opción **-subj "/CN=bob@example.com"**, se define el sujeto del certificado, el nombre común del mismo. Es el campo que se utiliza para identificar al titular del ceertificado (en este caso, Bob), y es un atrbuto importante que aparece en el certificado.

Validación del certificado

Una vez que ya se ha firmado la CSR, es decir, se ha emitido el certificado digital, Alice comprueba que el **certificado es válido**, según la cadena de confianza establecida por la CA. ¿Qué es una cadena de confianza?:

> Una cadena de confianza es un archivo que incluye los certificados de la CA intermedia y la CA raíz (la cadena de certificación completa). Estos certificados son los que se utilizan para validar el certificado de Bob. Es, en definitiva, la que permite verificar si el certificado de Bob fue emitido por una CA confiable.

> Recordemos que un certificado contiene información de la entidad a la qu representa y la clave pública, entre otras cosas.



Para ello, usamos la sentencia:

```
openssl verify -CAfile intermediate/certs/ca-chain.cert.pem
intermediate/certs/bob@example.com.cert.pem
```

La opción **-CAfile intermediate/certs/ca-chain.cert.pem**, especifica el archivo que contiene la cadena de confianza, e **intermediate/certs/bob@example.com.cert.pem** indica el nombre del certificado de Bob que se quiere validar.

El proceso de validación consiste en lo siguiente:

- **Comprobamos la firma**: se verifica que el certificado de Bob fue firmado por la CA intermedia. La CA intermedia firmó el certificado de Bob con su clave privada, y esta firma es ahora validada utilizando la clave pública de dicha CA, que está en la cadena de confianza proporcionada. También se comprueba que la CA intermedia fue firmada por la CA raíz, lo cual se hace también con la clave pública de la CA raíz, que está también en la cadena de confianza. También verifica si es válido en cuanto a la fecha de expiración.

Con esto, Alice habría verificado que el certificado de Bob fue emitido por una CA de confianza (en este caso, la intermedia, que es la que se encarga de ello, porque la raíz hemos dicho que solo se usa para crear intermedias), que es íntegro y que es válido en cuanto a fechas.

Alice, seguidamente, envía el certificado firmado a Bob, el cual instala en su navegador web y ahora es capaz de acceder a las imágenes del gatito de Alice.

Revocación del certificado

Tristemente resulta que Bob se está portando mal. Bob ha publicado imágenes del gatito de Alice en Hacker News, afirmando que son propias y ganando gran popularidad. Alice lo descubre y necesita revocar su acceso inmediato. Para ello, ejecuta la siguiente orden:

```
openssl ca -config intermediate/openssl.cnf -revoke
intermediate/certs/bob@example.com.cert.pem
```

El comando **openssl ca -revoke**, marca el certificado de Bob como revocado en la base de datos de la CA intermedia (intermediate/index.txt). Internamente, esto actualiza la lista de certificados gestionados por la CA con el estado del certificado de Bob, indicando que ya no es válido. De esta forma, Alice corta inmediatamente el acceso a Bob a su contenido protegido.

Aunque la revocación se registra en la base de datos interna de la CA intermedia, para que otros sistemas (otros navegadores o servidores) lo reconozcan o la autoridad necesite constatar que dicho certificado se ha revocado, en un momento posterior a la revocación, es necesario publicar una **CRL (Certificate Revocation List)**, que contendrá un listado de todos los certificados que han sido revocados. Alice puede crear la lista de la siguiente forma:

```
openssl ca -config intermediate/openssl.cnf -gencrl -out
intermediate/crl/intermediate.crl.pem
```

Aunque ya se ha puesto el certificado de Bob como revocado en la base de datos, y se ha constatado en la CRL, se puede verificar si el certificado sigue siendo válido teniendo en cuenta tanto la cadena de confianza como la lista de revocación de certificados (CRL):

```
openssl verify -CAfile intermediate/certs/ca-chain.cert.pem -CRLfile
intermediate/crl/intermediate.crl.pem -crl_check
intermediate/certs/bob@example.com.cert.pem
```

Con esta sentencia se vuelve a comprobar que el certificado de Bob fue emitido por una CA confiable (con **-CAfile intermediate/certs/ca-chain.cert.pem**), si aparece en la lista de revocación de certificados (CRL) (con la opción **-CRLfile intermediate/crl/intermediate.crl.pem**), y si está presente en la CRL y si ha sido revocado (con **-crl_check**). De esta forma:

- Si el certificado no está revocado y la cadena de confianza es válida, el comando confirmará que el certificado es válido.
- Si el certificado de Bob está en la CRL, el comando devolverá un error indicando que el certificado no es válido.

En caso de que, efectivamente, el certificado se encuentre en la lista de certificados revocados, se mostraría lo siguiente:

```
CN = bob@example.com

error 23 at 0 depth lookup: certificate revoked error
```

> intermediate/certs/bob@example.com.cert.pem: verification failed

El error 23 es un código de error de Openssl que indica un problema relacionado con la revocación de un certificado, y "at 0 depth lookup", significa que el error ocurrió en el certificado que se está verificando directamente, no en algún certificado intermedio de la cadena.