Firmar certificados de servidor y cliente

Suscribiremos certificados utilizando nuestra CA intermedia. Puede utilizar estos certificados firmados en una variedad de situaciones, como por ejemplo para proteger las conexiones a un servidor web o para autenticar clientes que se conectan a un servicio.

```
-----------------------------------------------------------------
(¡!) Nota

Los siguientes pasos son desde su punto de vista como la autoridad de
certificación. Un tercero, sin embargo, en su lugar puede crear su
propia solicitud de clave privada y el certificado de firma (CSR) sin
revelar su clave privada a usted. Ellos le darán su CSR y usted les
devolverá un certificado firmado. En ese escenario, omita las órdenes
genrsa and req.
-----------------------------------------------------------------
```

Crear una llave

Nuestros pares `root` e intermedios son 4096 bits. Los certificados de servidor y cliente normalmente expiran después de un año, por lo que se pueden utilizar con seguridad en su lugar 2048 bits.

```
(¡!) Nota

A pesar de que 4096 bits es un poco más seguro que 2048 bits, ello
ralentiza los "apretones de manos" TLS y aumenta significativamente la carga
del procesador durante uno de ellos. Por esta razón, la mayoría
de los sitios web utilizan pares de 2048 bits.
```

Si va a crear un par criptográfico para su uso con un servidor web (por ejemplo, Apache), tendrá que introducir la contraseña cada vez que se reinicia el servidor web. Es posible que desee omitir la opción `-aes256` para crear una clave sin contraseña.

```
# cd /root/ca
# openssl genrsa -aes256 \
      -out intermediate/private/www.example.com.key.pem 2048
# chmod 400 intermediate/private/www.example.com.key.pem
```

Crear un certificado

Use la clave privada para crear una solicitud de firma de certificado (CSR). Los detalles de CSR necesitan no coincidir con la CA intermedia. Para los certificados de servidor, el nombre común debe ser un nombre de dominio completo (por ejemplo, `www.example.com`), mientras que para los certificados de cliente puede ser cualquier identificador único (por ejemplo, una dirección de correo electrónico). Tenga en cuenta que el nombre común no puede ser el mismo que o bien el certificado de raíz o intermedia.

```
# cd /root/ca
# openssl req -config intermediate/openssl.cnf \
      -key intermediate/private/www.example.com.key.pem \
      -new -sha256 -out intermediate/csr/www.example.com.csr.pem

Enter pass phrase for www.example.com.key.pem: secretpassword
You are about to be asked to enter information that will be incorporated
into your certificate request.
-----
Country Name (2 letter code) [XX]:US
State or Province Name []:California
Locality Name []:Mountain View
Organization Name []:Alice Ltd
Organizational Unit Name []:Alice Ltd Web Services
Common Name []:www.example.com
Email Address []:
```

Para crear un certificado, utilice la CA intermedia al objeto de firmar la CSR. Si el certificado va a ser utilizado en un servidor, utilice la extensión `server_cert`. Si el certificado se va a utilizar para la autenticación de usuario, utilice la extensión `usr_cert`. A los certificados se les da generalmente una validez de un año, a pesar de que las CA suelen dar unos días extra para mayor comodidad.

```
# cd /root/ca
# openssl ca -config intermediate/openssl.cnf \
      -extensions server_cert -days 375 -notext -md sha256 \
      -in intermediate/csr/www.example.com.csr.pem \
      -out intermediate/certs/www.example.com.cert.pem \
      -subj "/CN=www.example.com"

Certificate is to be certified until Dec  8 21:55:22 2017 GMT (375 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

# chmod 444 intermediate/certs/www.example.com.cert.pem
```

El archivo `intermediate/index.txt` debería contener una línea haciendo referencia a este nuevo certificado.

```
V 160420124233Z 1000 unknown ... /CN=www.example.com
```

Comprobar el certificado

```
# openssl x509 -noout -text \
      -in intermediate/certs/www.example.com.cert.pem
```

El **Emisor** es la entidad emisora intermedia. El **Suject** se refiere al propio certificado.

```
Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=GB, ST=England,
            O=Alice Ltd, OU=Alice Ltd Certificate Authority,
            CN=Alice Ltd Intermediate CA
    Validity
        Not Before: Apr 11 12:42:33 2015 GMT
        Not After : Apr 20 12:42:33 2016 GMT
    Subject: C=US, ST=California, L=Mountain View,
             O=Alice Ltd, OU=Alice Ltd Web Services,
             CN=www.example.com
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
```

La salida también mostrará las **extensiones X509v3**. Al crear el certificado, usted utilizó ya sea la extensión `server_cert` o `usr_cert`. Las opciones de la sección de configuración correspondiente se reflejarán en la salida.

```
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    Netscape Cert Type:
        SSL Server
    Netscape Comment:
        OpenSSL Generated Server Certificate
    X509v3 Subject Key Identifier:
        B1:B8:88:48:64:B7:45:52:21:CC:35:37:9E:24:50:EE:AD:58:02:B5
    X509v3 Authority Key Identifier:
        keyid:69:E8:EC:54:7F:25:23:60:E5:B6:E7:72:61:F1:D4:B9:21:D4:45:E9
        DirName:/C=GB/ST=England/O=Alice Ltd/OU=Alice Ltd Certificate
Authority/CN=Alice Ltd Root CA
        serial:10:00

    X509v3 Key Usage: critical
         Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication
```

Utilice el archivo de cadena de certificados CA que hemos creado previamente (`ca-chain.cert.pem`) para verificar que el nuevo certificado tiene una cadena válida de confianza.

```
# openssl verify -CAfile intermediate/certs/ca-chain.cert.pem \
      intermediate/certs/www.example.com.cert.pem

www.example.com.cert.pem: OK
```

Desplegar el certificado

Ahora puede implementar su nuevo certificado a un servidor, o distribuir el certificado a un cliente. Al desplegar una aplicación del servidor (por ejemplo, Apache), es necesario hacer los siguientes archivos disponibles:

- `ca-chain.cert.pem`
- `www.example.com.key.pem`
- `www.example.com.cert.pem`

Si se está suscribiendo un CSR de un tercero, no tiene acceso a su clave privada por lo que sólo necesita devorverles el archivo de cadena (`ca-chain.cert.pem`) y el certificado (`www.example.com.cert.pem`).