TEMA 2: Administración de credenciales de acceso a sistemas

Práctica 2.5: Creación de una PKI en Linux

Una CA es una entidad de confianza, responsable de emitir y revocar certificados digitales que se pueden emplear para la firma electrónica o para encriptar archivos.

Un posible empleo de los certificados emitidos por la CA es poder dar al usuario la veracidad y autenticidad sobre el sitio web que se visita para evitar ser víctima de phishing. Otros usos de los certificados pasan por encriptar conexiones o para configurar conexiones VPN.

Existen entidades CA de pago. En nuestro caso, emitiremos certificados autofirmados que podremos emplear en una VPN, servidor, etc.

La infraestructura de una CA se va a apoyar en la herramienta openssl.

Configuración de la CA

Cuando se actúa como CA hay que manejar parejas de claves privadas y públicas. Al crear la CA raíz, esta pareja se convierte en la pareja raíz de claves.

En términos más precisos:

- Crearemos la clave privada, archivo ca.key.pem
- Certificado público, archivo ca.cert.pem

Por cuestiones de seguridad la CA raíz no firma certificados de servidores o clientes directamente. Emplearemos la CA raíz para crear otras CA subordinadas a las que les otorgaremos el privilegio de firmar certificados. De esta forma, podremos mantener segura la CA raíz.

Para almacenar todas las claves y certificados vamos a crear un directorio especifico con el comando:

```
mkdir -p /root/ca
```

Y dentro de él la estructura de subdirectorios:

```
mkdir certs crl newcerts private
```

Aseguraremos el directorio private con:

```
chmod 700 private
```

Crearemos los archivos “index.txt” y “serial” para almacenar la pista de los certificados que firmamos:

```
touch index.txt
echo 1000 > serial
```

Y para finalizar, crearemos el archivo /root/ca/openssl.cnf con el siguiente contenido

https://drive.google.com/file/d/1MA3iipm5BuepegpItceyllwGTdQXciYF/view?usp=sharing

Creación de la clave privada de la CA raíz

Procederemos a crear el archivo “ca.key.pem”. Es la clave privada de la CA raíz. Cualquier sujeto en posesión de la clave podrá emitir certificados como si fuera el usuario legítimo.

Para aumentar la seguridad, emplearemos claves de 4096 bits en todas nuestras operaciones.

Nos situaremos en el directorio /root/ca con el comando:

cd /root/ca

La creación de la clave privada emplea protección por contraseña, utilizaremos “123456”.

Ejecutaremos el comando:

root@ubuntu20:~/ca# openssl genrsa -aes256 -out private/ca.key.pem 4096

```
root@ubuntu20:~/ca# openssl genrsa -aes256 -out private/ca.key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....................................................................++++
.....................................................................++++
e is 65537 (0x010001)
Enter pass phrase for private/ca.key.pem:
Verifying - Enter pass phrase for private/ca.key.pem:
```

Protegeremos el archivo, asegurando los privilegios:

root@ubuntu20:~/ca# chmod 400 private/ca.key.pem

Creación del certificado raíz CA raíz

Emplearemos la anterior clave privada generada para crear el certificado raíz “ca.cert.pem”. Daremos un valor de caducidad alto ya que cuando el certificado raíz caduca, todo lo firmado por la CA se invalida.

Ejecutamos el comando:

root@ubuntu20:~/ca# openssl req -key private/ca.key.pem -config /root/ca/openssl.cnf -new -x509 -days 7300 -extensions v3_ca -out certs/ca.cert.pem

Nos pedirá la contraseña para introducida anteriormente: 123456

```
root@ubuntu20:~/ca# openssl req -key private/ca.key.pem -new -x509 -days 7300 -extensions v3_ca -out certs/ca.cert.pem
Enter pass phrase for private/ca.key.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) [ES]:
State or Province Name (full name) [Spain]:
Locality Name (eg, city) []:Albacete
Organization Name (eg, company) [Mi empresa, S.A.]:
Organizational Unit Name (eg, section) []:Departamento de ejemplo
Common Name (e.g. server FQDN or YOUR name) []:MI-CA
Email Address []:mica@miempresa.com
```

Protegeremos el archivo creado con el comando:

root@ubuntu20:~/ca# chmod 444 certs/ca.cert.pem

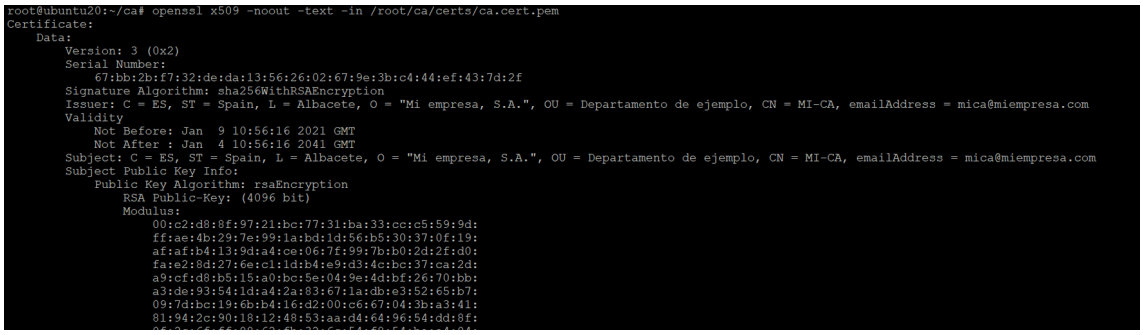
Verificación del certificado raíz de la CA

El certificado raíz que hemos creado está en /root/ca/certs/ca.cert.pem. Vamos a comprobar que el certificado es autofirmado por que coincide el campo “Issuer” con el campo “Subject”.

Ejecutaremos el comando:

root@ubuntu20:~/ca# openssl x509 -noout -text -in /root/ca/certs/ca.cert.pem

Y obtenemos la salida



Hemos obtenido información sobre:

- Algoritmo empleado en el campo “Signature Algorithm”.
- Caducidad, en el campo “Validity”.
- Longitud de la clave pública, “Public-Key”.
- Campo “Issuer”, quien firma el certificado.
- Campo “Subject”, a quien se refiere el certificado.

Creación de CA intermedias

Son CA que pueden firmar certificados en nombre de la CA raíz. Principalmente emplearemos CA intermedias por seguridad. La clave privada raíz debe permanecer oculta. Emplearemos la clave intermedia y si esta se compromete, se revocarán todos los certificados emitidos por la CA intermedia.

Todos los archivos que creemos con la CA intermedia se almacenarán en /root/ca/intermedia.

Crearemos el directorio con la misma estructura que la CA raíz y protegeremos el directorio private.

```
root@ubuntu20:~/ca# mkdir -p /root/ca/intermedia

root@ubuntu20:~/ca# cd /root/ca/intermedia/

root@ubuntu20:~/ca/intermedia# mkdir certs crl csr newcerts private

root@ubuntu20:~/ca/intermedia# chmod 700 private
```

Creamos los archivos “index.txt” y “serial”, además del archivo “crlnumber” para llevar una traza de las listas de certificados revocados

```
root@ubuntu20:~/ca/intermedia# touch index.txt

root@ubuntu20:~/ca/intermedia# echo 1000 > serial

root@ubuntu20:~/ca/intermedia# echo 1000 > crlnumber
```

Copiaremos el archivo /root/ca/openssl.cnf al directorio /root/ca/intermedia

```
root@ubuntu20:~/ca/intermedia# cp /root/ca/openssl.cnf /root/ca/intermedia/
```

Ajustaremos la sección de directorios del archivo cambiando los valores:

- dir
- private_key
- certificate
- crl

- policy = policy_loose

```
[ CA_default ]

dir             = /root/ca/intermedia           # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
#unique_subject = no                    # Set to 'no' to allow creation of
                                        # several certs with same subject.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/intermedia.cert.pem      # The CA certificate
serial          = $dir/serial           # The current serial number
crlnumber       = $dir/crlnumber        # the current crl number
                                        # must be commented out to leave a V1 CRL
crl             = $dir/intermedia.crl.pem               # The current CRL
private_key     = $dir/private/intermedia.key.pem# The private key

x509_extensions = usr_cert              # The extensions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt        = ca_default            # Subject Name options
cert_opt        = ca_default            # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions        = crl_ext
crl_extensions  = crl_ext
default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = sha256                # use public key default MD
preserve        = no                    # keep passed DN ordering
```

Creación de la clave intermedia

Crearemos la clave privada para la CA intermedia. Se aconseja encriptar la clave, emplear una contraseña robusta y almacenarla en un lugar seguro.

Nos cambiamos al directorio /root/ca y ejecutamos el siguiente comando:

root@ubuntu20:~/ca# openssl genrsa -aes256 -out intermedia/private/intermedia.key.pem 4096

```
root@ubuntu20:~/ca# openssl genrsa -aes256 -out intermedia/private/intermedia.key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....................................++++
.....................................................................................++++
e is 65537 (0x010001)
Enter pass phrase for intermedia/private/intermedia.key.pem:
Verifying - Enter pass phrase for intermedia/private/intermedia.key.pem:
```

Protegemos la clave, cambiando los permisos con el comando:

root@ubuntu20:~/ca# chmod 400 intermedia/private/intermedia.key.pem

Creación del certificado intermedio

Crearemos el certificado intermedio empleando la clave intermedia generada anteriormente. El archivo a generar será “intermedia.csr.pem”. En este caso generamos una petición de certificado y el valor del campo “Common NAme” debe de ser diferente.

Nos situaremos en el directorio /root/ca y generaremos la petición con el comando:

root@ubuntu20:~/ca# openssl req -config intermedia/openssl.cnf -new -sha256 -key intermedia/private/intermedia.key.pem -out intermedia/csr/intermedia.csr.pem

```
root@ubuntu20:~/ca# openssl req -config intermedia/openssl.cnf -new -sha256 -key intermedia/private/intermedia.ke
Enter pass phrase for intermedia/private/intermedia.key.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) [ES]:
State or Province Name (full name) [Spain]:
Locality Name (eg, city) []:Albacete
Organization Name (eg, company) [Mi empresa, S.A.]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Mi-CA Intermedia
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@ubuntu20:~/ca#
```

Crearemos ahora el certificado de la CA intermedia empleando el archivo de configuración de la CA raíz (/root/ca/openssl.cnf) junto con su certificado público firmando la petición que hemos generado antes (intermedia.csr.pem) y generaremos el archivo intermedia.cert.pem

root@ubuntu20:~/ca# openssl ca -config openssl.cnf -extensions v3_ca_intermediate -days 3650 -notext -md sha256 -in intermedia/csr/intermedia.csr.pem -out intermedia/certs/intermedia.cert.pem

```
root@ubuntu20:~/ca# openssl ca -config openssl.cnf -extensions v3_ca_intermediate -days 3650 -notext -md sha256 -in intermedia/csr/intermedia.cst.pem -out in
termedia/certs/intermedia.cert.pem
Using configuration from openssl.cnf
Enter pass phrase for /root/ca/private/ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Jan  9 11:31:06 2021 GMT
            Not After : Jan  7 11:31:06 2031 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Spain
            organizationName          = Mi empresa, S.A.
            commonName                = Mi-CA Intermedia
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                B6:6E:4D:C4:C5:AD:5D:2C:F9:8A:A3:E0:10:C9:C8:FF:A5:6E:7B:2B
            X509v3 Authority Key Identifier:
                keyid:8D:C6:36:43:56:D7:1F:19:B2:FB:1D:3B:32:8C:D1:D3:CF:1E:CF:51

            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
Certificate is to be certified until Jan  7 11:31:06 2031 GMT (3650 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
```

Protegeremos el archivo creado con el comando:

root@ubuntu20:~/ca# chmod 444 intermedia/certs/intermedia.cert.pem

Si miramos ahora el archivo “index.txt” vemos donde OpenSSL ha almacenado lo que acabamos de hacer

```
root@ubuntu20:~/ca# cat index.txt
V       310107113106Z           1000    unknown /C=ES/ST=Spain/O=Mi empresa, S.A./CN=Mi-CA Intermedia
root@ubuntu20:~/ca#
```

Verificación del certificado intermedio

Comprobaremos los detalles del certificado creado para la CA intermedia. Nos situaremos en el directorio /root/ca y ejecutaremos:

root@ubuntu20:~/ca# openssl x509 -noout -text -in intermedia/certs/intermedia.cert.pem

Y lo podemos verificar contra el certificado raíz para comprobar la cadena de confianza:

root@ubuntu20:~/ca# openssl verify -CAfile certs/ca.cert.pem intermedia/certs/intermedia.cert.pem

```
root@ubuntu20:~/ca# openssl verify -CAfile certs/ca.cert.pem intermedia/certs/intermedia.cert.pem
intermedia/certs/intermedia.cert.pem: OK
```

Concatenación de certificados

Cuando una aplicación intenta verificar un certificado firmado por una CA intermedia, debe verificar también el certificado intermedio contra el certificado raíz. Para la cadena de confianza, crearemos un certificado cadena para las aplicaciones.

Nos situaremos en el directorio /root/ca y juntamos los dos certificados

root@ubuntu20:~/ca# cat intermedia/certs/intermedia.cert.pem certs/ca.cert.pem >intermedia/certs/ca-cadena.cert.pem

Protegeremos el archivo

Firmar certificados de servidor y cliente.

Realizaremos el proceso de firma empleando la CA intermedia. Los certificados que creemos, se pueden utilizar para un servidor web o autenticar clientes en un servicio entre otras situaciones.

Supongamos que tenemos un cliente que solicita un certificado para un dominio llamado “midominio.com”.

Cuando se solicita un certificado se necesita una clave asociada al cliente para empezar a firmar todo el proceso. La clave la puede proporcionar el cliente o a través de un intermediario.

Para simplificar el paso, generaremos la clave del cliente para poder iniciar el proceso.

Se genera una petición de certificado CSR que luego se firma obteniendo el certificado para el usuario.

Ejecutaremos el comando que genera la firma para el cliente:

root@ubuntu20:~/ca# openssl genrsa -out intermedia/private/midominio.key.pem 2048

```
root@ubuntu20:~/ca# openssl genrsa -out intermedia/private/midominio.key.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................................................
.............+++++
e is 65537 (0x010001)
root@ubuntu20:~/ca#
```

Aseguramos el archivo

root@ubuntu20:~/ca# chmod 400 intermedia/private/midominio.key.pem

Con la clave vamos a crear ahora la petición de certificado CSR. Los datos con los que vamos a rellenar la solicitud son relativos al cliente.

root@ubuntu20:~/ca# openssl req -config intermedia/openssl.cnf -key intermedia/private/midominio.key.pem -new -sha256 -out intermedia/csr/midominio.csr.pem

```
root@ubuntu20:~/ca# openssl req -config intermedia/openssl.cnf -key intermedia/private/midominio.key.pem -new -sha256 -out intermedia/csr/midominio.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) [ES]:
State or Province Name (full name) [Spain]:
Locality Name (eg, city) []:Albacete
Organization Name (eg, company) [Mi empresa, S.A.]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:midominio.com
Email Address []:admin@midominio.com

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

Ahora firmaremos la CSR para obtener el certificado. Si es un certificado para servidores emplearemos el parámetro “-extensions server_cert”, si es para un usuario “-extensions usr_cert”. La validez suele ser 1 año.

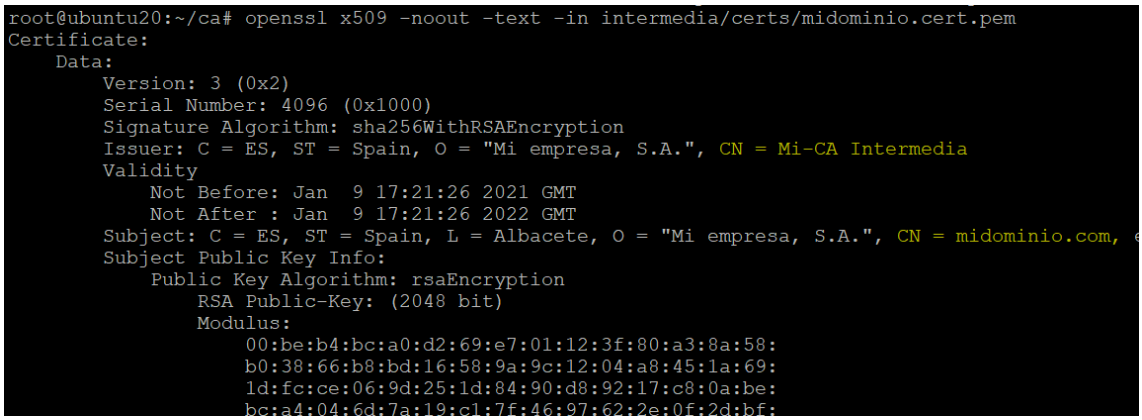
root@ubuntu20:~/ca# openssl ca -config intermedia/openssl.cnf -extensions server_cert -days 365 -notext -md sha256 -in intermedia/csr/midominio.csr.pem -out intermedia/certs/midominio.cert.pem

```
root@ubuntu20:~/ca# openssl ca -config intermedia/openssl.cnf -extensions server_cert
intermedia/certs/midominio.cert.pem
Using configuration from intermedia/openssl.cnf
Enter pass phrase for /root/ca/intermedia/private/intermedia.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Jan  9 17:21:26 2021 GMT
            Not After : Jan  9 17:21:26 2022 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Spain
            localityName              = Albacete
            organizationName          = Mi empresa, S.A.
            commonName                = midominio.com
            emailAddress              = admin@midominio.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Cert Type:
                SSL Server
            Netscape Comment:
                Certificado para servidores generado por la CA
            X509v3 Subject Key Identifier:
                D7:62:99:34:77:1A:46:2A:38:0C:18:C6:D7:91:70:D7:CB:D4:61:24
            X509v3 Authority Key Identifier:
                keyid:B6:6E:4D:C4:C5:AD:5D:2C:F9:8A:A3:E0:10:C9:C8:FF:A5:6E:7B:2B
                DirName:/C=ES/ST=Spain/L=Albacete/O=Mi empresa, S.A./OU=Departamento
                serial:10:00

            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
Certificate is to be certified until Jan  9 17:21:26 2022 GMT (365 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
root@ubuntu20:~/ca#
```

Si comprobamos el contenido del archivo index.txt de la CA intermedia veremos que ha actualizado con lo que acabamos de realizar

```
root@ubuntu20:~/ca# cat intermedia/index.txt
V       220109172126Z           1000    unknown /C=ES/ST=Spain/L=Albacete/O=Mi empresa, S.A./CN=midominio.com/emailAddress=admin@midominio.com
root@ubuntu20:~/ca#
```

Verificamos el certificado obtenido para comprobar que el “Issuer” es la CA intermedia y “Subject” es “midominio.com”.



Comprobamos también la cadena de confianza con el siguiente comando:

root@ubuntu20:~/ca# openssl verify -CAfile intermedia/certs/ca-cadena.cert.pem intermedia/certs/midominio.cert.pem

Cuando hagamos uso de este certificado para un servidor web, hemos de emplear tres archivos:

- ca-cadena.cert.pem
- midominio.key.pem
- midominio.cert.pem

Listas de revocación de certificados

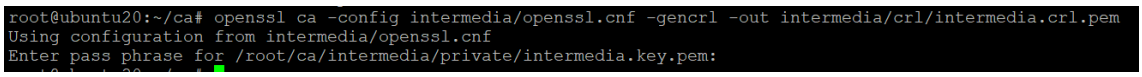
Una lista de revocación de certificados CRL es una lista donde se almacenan los certificados que han sido revocados por la CA. Cuando un navegador web intenta comprobar la autenticidad del servidor consulta la lista para determinar si está revocado o no. En el caso de una conexión VPN, emplear un certificado revocado implica la pérdida del servicio.

La lista se suele publicar en una url pública y sencilla, por ejemplo http://mica.com/intermedia.crl.pem

Cuando una CA firma un certificado, codifica la url de la lista dentro del propio certificado. Se puede configurar con el parámetro “crlDistributionPoints” en la sección “server_cert” del archivo de configuración openssl.cnf.

Crearemos la crl para la CA intermedia. Ejecutaremos el siguiente comando:

root@ubuntu20:~/ca# openssl ca -config intermedia/openssl.cnf -gencrl -out intermedia/crl/intermedia.crl.pem



Comprobaremos el estado de la lista de certificados revocados con el comando:

root@ubuntu20:~/ca# openssl crl -in intermedia/crl/intermedia.crl.pem -noout -text

Vemos que nos dirá que no hay ningún certificado revocado.

```
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = ES, ST = Spain, O = "Mi empresa, S.A.", CN = Mi-CA Intermedia
        Last Update: Jan  9 17:31:52 2021 GMT
        Next Update: Feb  8 17:31:52 2021 GMT
        CRL extensions:
            X509v3 Authority Key Identifier:
                keyid:B6:6E:4D:C4:C5:AD:5D:2C:F9:8A:A3:E0:10:C9:C8:FF:A5:6E:7B:2B

            X509v3 CRL Number:
                4096
No Revoked Certificates.
```

Vamos a revocar el certificado generado anteriormente para midominio.com

root@ubuntu20:~/ca# openssl ca -config intermedia/openssl.cnf -revoke intermedia/certs/midominio.cert.pem

```
root@ubuntu20:~/ca# openssl ca -config intermedia/openssl.cnf -revoke intermedia/certs/midominio.cert.pem
Using configuration from intermedia/openssl.cnf
Enter pass phrase for /root/ca/intermedia/private/intermedia.key.pem:
Revoking Certificate 1000.
Data Base Updated
```

Comprobamos el archivo “index.txt” de la entidad CA intermedia

```
root@ubuntu20:~/ca# cat intermedia/index.txt
R       210109172126Z   210109173500Z   1000    unknown /C=ES/ST=Spain/L=Albacete/O=Mi empresa, S.A./CN=midominio.com/
root@ubuntu20:~/ca#
```

Vemos que ahora lleva delante una R.
Ahora recreamos la lista CRL y la comprobamos.

root@ubuntu20:~/ca# openssl ca -config intermedia/openssl.cnf -gencrl -out intermedia/crl/intermedia.crl.pem

root@ubuntu20:~/ca# openssl crl -in intermedia/crl/intermedia.crl.pem

```
root@ubuntu20:~/ca# openssl crl -in intermedia/crl/intermedia.crl.pem
-----BEGIN X509 CRL-----
MIIC5TCBzgIBATANBgkqhkiG9w0BAQsFADBTMQswCQYDVQQGEwJFUzEOMAwGA1UE
CAwFU3BhaW4xGTAXBgNVBAoMEE1pIGVtcHJlc2EsIFMuQS4xGTAXBgNVBAMMEE1p
LUNBIEludGVybWVkaWEXDTIxMDEwOTE3Mzc1OFoXDTIxMDIwODE3Mzc1OFowFTAT
AgIQABcNMjEwMTA5MTczNTAwWqAwMC4wHwYDVR0jBBgwFoAUtm5NxMWtXSz5iqPg
EMnI/6VueyswCwYDVR0UBAQCAhABMA0GCSqGSIb3DQEBCwUAA4ICAQCf9S6l/LyU
EtW0fVm4fTEo9reeTtzD3fa3iXqZqcjuX6RpPgnze8CE9q9DdnYFs5pHE+juQCrF
eade/c+1zJ1q6kMRIhQRu0f+qmmoTU32JHJ0w4VINiY3cJIqHCMvmsrgLKyNyQ8y
xd5VtQwIJGyqtPkEQWXSGrjIhqgmOIzbodLM0/EeIoBHTIFXlJe4vM6raX0ZMC+0
vaNAgfS8qzhLC8sUEnPViUojPdNAhN6VnUzdYeAZ3aYAm1S2RbLwjeTUGT80dQt6
xCSTDrWt1aMyaIqGi4MJFa1EDgg3xpc8yl7QZrh85fUwj5BHEy72v6fmU3T498H5
6V/Yhd3FkN1+HoQW4iXhFnk/XwL0LyZg3xVqms7mnf2LZZEu98LW25sqLS4E68Xv
lMe2XTeCREU4+Odv2992/1XVIlaAgHsZJE3KSmxa4jfBYyYS2GLaaaH6GG61j5aR
Dju+zx7t5ARwlEhFmsT8BHVuEMSc7I1INLzYsHMsxI5KGc/gsaC+yKr7T8RYXcCY
2vOf0oTzbW25zJ/mnFsWDPktIVVPCm/V9ZNqp9ZIwymY+zmaCZ+F6VGi1mlCJvpE
OVnKC7R9cbo61Drd8VoZwoS8iWEPjvtpvDB/a0K4p+X//unwpQBcsA+mLVVYNecX
Ju1fvvilAVUEnUNBc5dig6deqbvnexGirA==
-----END X509 CRL-----
```