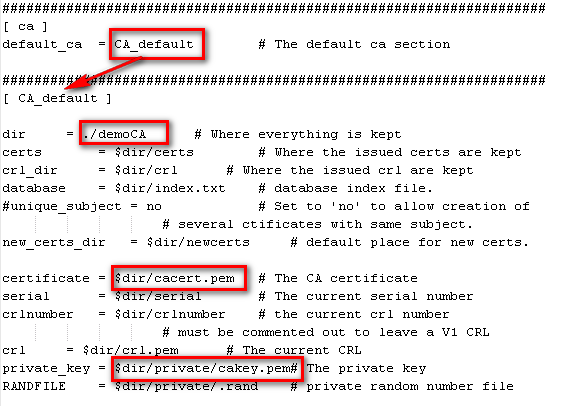
**Using OpenSSL to create your own CA***JeremyC 20/2/2016*Based on the following*:*[*https://jamielinux.com/docs/openssl-certificate-authority/index.html*](https://jamielinux.com/docs/openssl-certificate-authority/index.html)

**NOTE:***Typically, the root CA does not sign server or client certificates directly. The root CA is only ever used to create one or more intermediate CAs, which are trusted by the root CA to sign certificates on their behalf. This is best practice. It allows the root key to be kept offline and unused as much as possible, as any compromise of the root key is disastrous.*

**Creating a root CA key pair (cakey.pem and cacert.pem)**o Create a directory to store all keys and certificates:  
mkdir c:\ca  
  
o Create the following structure:  
C:\ca>mkdir certs crl newcerts private && type nul >>index.txt && echo 1000 > serial  
  
o Copy over the openssl.cnf file.  
C:\ca>copy c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\openssl.cnf .

1 file(s) copied.  
  
Note the following values in openssl.cnf:  


**Note:** I added the following section for an intermediate CA created later (with cert that is signed by the root CA):  
…

# You can even override a supported extension:

# basicConstraints= critical, DER:30:03:01:01:FF

**[ v3\_intermediate\_ca ]**

**# Extensions for a typical intermediate CA (`man x509v3\_config`).**

**subjectKeyIdentifier = hash**

**authorityKeyIdentifier = keyid:always,issuer**

**basicConstraints = critical, CA:true, pathlen:0**

**keyUsage = critical, digitalSignature, cRLSign, keyCertSign**

[ crl\_ext ]

# CRL extensions.

# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy

…

o Create the root CA key  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe genrsa -aes256 -out private/cakey.pem 4096

Loading 'screen' into random state - done

Generating RSA private key, 4096 bit long modulus

....................................................................................................

.............................................................................................++

....++

e is 65537 (0x10001)

Enter pass phrase for private/cakey.pem:

Verifying - Enter pass phrase for private/cakey.pem:  
  
**NOTE**: This command does not seem to use the openssl.cnf file. I renamed the various openssl.cnf files I could find, and this command still worked.  
  
  
o Create the root CA cert  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe req -config c:\ca\openssl.cnf -key private/cakey.pem -new -x509 -days 7300 –sha256 -extensions v3\_ca -out certs/cacert.pem

Enter pass phrase for private/cakey.pem:

Loading 'screen' into random state - done

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]:GB

State or Province Name (full name) [Some-State]:England

Locality Name (eg, city) []:

Organization Name (eg, company) [Internet Widgits Pty Ltd]:JeremyC

Organizational Unit Name (eg, section) []:JeremyC CA

Common Name (eg, YOUR name) []:JeremyC Root CA

Email Address []:

o Verify the root CA cert  
C:\ca>c:\users\administrator.CWLAB\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe x509 -noout -text -in certs/cacert.pem

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

c1:40:01:d4:46:17:21:dc

Signature Algorithm: sha256WithRSAEncryption

Issuer: C=GB, ST=England, O=JeremyC, OU=JeremyC CA, CN=JeremyC Root CA

Validity

Not Before: Feb 20 11:46:16 2016 GMT

Not After : Feb 15 11:46:16 2036 GMT

Subject: C=GB, ST=England, O=JeremyC, OU=JeremyC CA, CN=JeremyC Root CA

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (4096 bit)

Modulus (4096 bit):

00:c6:43:db:41:7c:57:b5:cb:5f:0a:75:7e:c6:de:

f5:6f:b6:bf:7c:e3:f3:03:ea:cb:ca:af:49:1b:dd:

ec:6c:ee:d0:c1:8b:3a:58:6e:06:df:4d:6d:d9:e1:

d1:df:dd:2a:69:68:d9:7f:0d:34:ea:04:d8:3b:c6:

ed:90:32:23:f8:ed:d4:84:74:e8:f4:84:f9:80:f9:

46:1e:24:cc:7f:bf:55:93:b4:5b:99:f7:4a:d6:6e:  
…  
  
  
**Create the intermediate CA key pair**

*“An intermediate certificate authority (CA) is an entity that can sign certificates on behalf of the root CA. The root CA signs the intermediate certificate, forming a chain of trust. The purpose of using an intermediate CA is primarily for security. The root key can be kept offline and used as infrequently as possible. If the intermediate key is compromised, the root CA can revoke the intermediate certificate and create a new intermediate cryptographic pair.”*  
  
o Create folder structure  
mkdir c:\ca\intermediate  
cd intermediate  
mkdir certs crl csr newcerts private  
type nul >>index.txt (NOTE: This creates an empty file)  
echo 1000 > serial  
echo 1000 > crlnumber (NOTE: To keep track of certificate revocation lists)  
  
o Use a copy of the same openssl.cnf file that was used for the root cert, but with the following changes:  
[ CA\_default ]

dir = c:/ca/intermediate

private\_key = $dir/private/intermediate.key.pem

certificate = $dir/**certs**/intermediate.cert.pem

crl = $dir/**crl**/intermediate.crl.pem

policy = policy\_anything  
  
  
o Create the intermediate CA key  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe genrsa -aes256 -out intermediate/private/intermediate.key.pem 4096

Loading 'screen' into random state - done

Generating RSA private key, 4096 bit long modulus

....................................................................................................

..........++

....................................................................................................

....................................................................................................

......................................................++

e is 65537 (0x10001)

Enter pass phrase for intermediate/private/intermediate.key.pem:

Verifying - Enter pass phrase for intermediate/private/intermediate.key.pem:

o Create a CSR for the intermediate CA cert  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe req -config c:/ca/intermediate/openssl.cnf -new -sha256 -key intermediate/private/intermediate.key.pem -out intermediate/csr/intermediate.csr.pem

Enter pass phrase for intermediate/private/intermediate.key.pem:

Loading 'screen' into random state - done

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]:GB

State or Province Name (full name) [Some-State]:England

Locality Name (eg, city) []:

Organization Name (eg, company) [Internet Widgits Pty Ltd]:JeremyC

Organizational Unit Name (eg, section) []:JeremyC CA

Common Name (eg, YOUR name) []:JeremyC Intermediate CA

Email Address []:

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:123456

An optional company name []:  
  
  
o Use the root CA to sign the intermediate CSR, thereby creating the intermediate CA cert  
We will make the intermediate cert valid for 10 years.  
**NOTE**: We need to point at the root CA openssl.cnf file.  
  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe ca -config c:/ca/openssl.cnf -extensions v3\_intermediate\_ca -days 3650 -notext -md sha256 -in intermediate/csr/intermediate.csr.pem -out intermediate/certs/intermediate.cert.pem

Using configuration from c:/ca/openssl.cnf

Loading 'screen' into random state - done

Enter pass phrase for c:/ca/private/cakey.pem:

Error opening CA certificate c:/ca/cacert.pem

4908:error:02001002:system library:fopen:No such file or directory:.\crypto\bio\bss\_file.c:356:fopen

('c:/ca/cacert.pem','rb')

4908:error:20074002:BIO routines:FILE\_CTRL:system lib:.\crypto\bio\bss\_file.c:358:

unable to load certificate

I had to do this to fix this…

C:\ca>copy certs\cacert.pem .

1 file(s) copied.  
  
It then worked…

C:\ca>c:\users\administrator.CWLAB\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe ca -config c:/ca/openssl.cnf -extensions v3\_intermediate\_ca -days 3650 –notext -md sha256 -in intermediate/csr/intermediate.csr.pem -out intermediate/certs/intermediate.cert.pem

Using configuration from c:/ca/openssl.cnf

Loading 'screen' into random state - done

Enter pass phrase for c:/ca/private/cakey.pem:

Check that the request matches the signature

Signature ok

Certificate Details:

Serial Number: 4096 (0x1000)

Validity

Not Before: Feb 20 13:14:31 2016 GMT

Not After : Feb 17 13:14:31 2026 GMT

Subject:

countryName = GB

stateOrProvinceName = England

organizationName = JeremyC

organizationalUnitName = JeremyC CA

commonName = JeremyC Intermediate CA

X509v3 extensions:

X509v3 Subject Key Identifier:

60:66:4E:74:48:61:65:9D:B1:8D:D5:2E:6A:80:BA:44:94:6D:5D:42

X509v3 Authority Key Identifier:

keyid:84:0F:52:F6:75:44:47:F2:AD:AC:11:A8:72:E3:08:CC:04:BE:E7:A7

X509v3 Basic Constraints: critical

CA:TRUE, pathlen:0

X509v3 Key Usage: critical

Digital Signature, Certificate Sign, CRL Sign

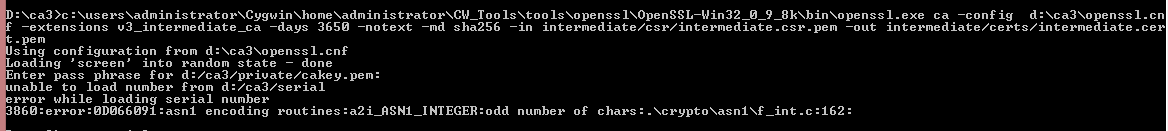
Certificate is to be certified until Feb 17 13:14:31 2026 GMT (3650 days)

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

**Note:** I got a strange error when I had “9999999” in my serial file:  
  
I fixed this by instead using “999999”.

**NOTE**: The “index.txt” file should now contain a reference to intermediate certificate:  
C:\ca>type index.txt

V 260217131431Z 1000 unknown /C=GB/ST=England/O=JeremyC/OU=JeremyC CA/CN=JeremyC Intermediate CA

o Verify the intermediate CA certificate  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe x509 -noout -text -in intermediate/certs/intermediate.cert.pem

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 4096 (0x1000)

Signature Algorithm: sha256WithRSAEncryption

Issuer: C=GB, ST=England, O=JeremyC, OU=JeremyC CA, CN=JeremyC Root CA

Validity

Not Before: Feb 20 13:14:31 2016 GMT

Not After : Feb 17 13:14:31 2026 GMT

Subject: C=GB, ST=England, O=JeremyC, OU=JeremyC CA, CN=JeremyC Intermediate CA

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (4096 bit)

Modulus (4096 bit):

00:e0:80:dc:82:20:37:0d:98:82:b5:06:0c:64:dc:

f1:bc:76:df:70:bf:61:2f:21:bd:ad:33:c0:90:de:  
…  
  
**Note:** In a later test, I used a root CA’s “serial” file containing “999999”. This ensured that the serial number given to my signed intermediate CA’s certificate was larger than “Serial Number: 4096 (0x1000)“:  
Serial Number: 10066329 (0x999999)  
  
o Verify the intermediate certificate against the root cert  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe verify -CAfile certs/cacert.pem intermediate/certs/intermediate.cert.pem

intermediate/certs/intermediate.cert.pem: OK  
  
**NOTE**: The client will now need the intermediate cert in order to validate the chain, plus the client will also need to install the root CA cert (or it could be bunded together with the intermediate cert: “cat intermediate/certs/intermediate.cert.pem certs/cacert.pem > intermediate/certs/ca-chain.cert.pem”).

**Creating an example certficiate (i.e. signed by out intermediate CA)***“We will be signing certificates using our intermediate CA.”*  
  
**NOTE**: “*Although 4096 bits is slightly more secure than 2048 bits, it slows down TLS handshakes and significantly increases processor load during handshakes. For this reason, most websites use 2048-bit pairs.*”

o Create a private key  
**NOTE**: The client would normally do this their end (e.g. in a Java keystore), and we wouldn’t see their private key.  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe genrsa -aes256 -out intermediate/private/www.example.com.key.pem 2048

Loading 'screen' into random state - done

Generating RSA private key, 2048 bit long modulus

.......................+++

...........................................+++

e is 65537 (0x10001)

Enter pass phrase for intermediate/private/www.example.com.key.pem:

Verifying - Enter pass phrase for intermediate/private/www.example.com.key.pem:  
  
**NOTE**: “*You may want to omit the -aes256 option to create a key without a password.”*  
  
  
o Use our new private key to create a CSR.   
**NOTE**: “*For server certificates, the****Common Name****must be a fully qualified domain name (eg, www.example.com), whereas for client certificates it can be any unique identifier (eg, an e-mail address). Note that the****Common Name****cannot be the same as either your root or intermediate certificate.”*  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe 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 intermediate/private/www.example.com.key.pem:

Loading 'screen' into random state - done

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]:GB

State or Province Name (full name) [Some-State]:England

Locality Name (eg, city) []:Berkshire

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Example COM Ltd

Organizational Unit Name (eg, section) []:Example COM Website

Common Name (eg, YOUR name) []:www.example.com

Email Address []:

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:  
  
  
**o Create our new cert, by using our intermediate CA to sign our CSR:**  
C:\ca\intermediate>type nul >>index.txt.attr

C:\ca\intermediate>cd ..  
  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe ca -config intermediate/openssl.cnf -extensions usr\_cert -days 375 -notext -md sha256 -in intermediate/csr/www.example.com.csr.pem -out intermediate/certs/www.example.com.cert.pem

Using configuration from intermediate/openssl.cnf

Loading 'screen' into random state - done

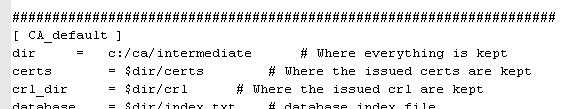
Enter pass phrase for c:/ca/intermediate/private/intermediate.key.pem:

Check that the request matches the signature

Signature ok

The organizationName field needed to be the same in the

CA certificate (JeremyC) and the request (Example COM Ltd)

**NOTE**: This is an error!   
  
This error occurred because my intermediate CA openssl.cnf file defined the following restriction:  
  
…  
  
  
  
To fix this, I changed it to this:  
  
  
  
The certificate was then signed successfully:  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe ca -config intermediate/openssl.cnf -extensions usr\_cert -days 375 -notext –md sha256 -in intermediate/csr/www.example.com.csr.pem -out intermediate/certs/www.example.com.cert.pem

Using configuration from intermediate/openssl.cnf

Loading 'screen' into random state - done

Enter pass phrase for c:/ca/intermediate/private/intermediate.key.pem:

Check that the request matches the signature

Signature ok

Certificate Details:

Serial Number: 4096 (0x1000)

Validity

Not Before: Feb 20 13:56:36 2016 GMT

Not After : Mar 1 13:56:36 2017 GMT

Subject:

countryName = GB

stateOrProvinceName = England

localityName = Berkshire

organizationName = Example COM Ltd

organizationalUnitName = Example COM Website

commonName = www.example.com

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

Netscape Comment:

OpenSSL Generated Certificate

X509v3 Subject Key Identifier:

45:A7:F7:A8:D7:EA:12:77:67:36:63:93:04:4D:49:4A:FC:EC:B8:43

X509v3 Authority Key Identifier:

keyid:60:66:4E:74:48:61:65:9D:B1:8D:D5:2E:6A:80:BA:44:94:6D:5D:42

Certificate is to be certified until Mar 1 13:56:36 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

o Verify the new certificate  
FYI: Before I had corrected the intermediate CA openssl.cnf file (see above):  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe x509 -noout -text -in intermediate/certs/www.example.com.cert.pem

unable to load certificate

604:error:0906D06C:PEM routines:PEM\_read\_bio:no start line:.\crypto\pem\pem\_lib.c:650:Expecting: TRU

STED CERTIFICATE  
  
After I had corrected the intermediate CA openssl.cnf file (see above) and therefore successfully signed the cert:  
C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe x509 -noout -text -in intermediate/certs/www.example.com.cert.pem

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 4096 (0x1000)

Signature Algorithm: sha256WithRSAEncryption

Issuer: C=GB, ST=England, O=JeremyC, OU=JeremyC CA, CN=JeremyC Intermediate CA

Validity

Not Before: Feb 20 13:56:36 2016 GMT

Not After : Mar 1 13:56:36 2017 GMT

Subject: C=GB, ST=England, L=Berkshire, O=Example COM Ltd, OU=Example COM Website, CN=www.ex

ample.com

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (2048 bit)

Modulus (2048 bit):

00:9b:1b:b6:7d:04:51:31:07:dc:fa:74:e9:e6:f3:  
…  
  
  
o Validate the cert (i.e. against the root CA cert + intermediate CA cert ‘bundle’)  
C:\ca>type intermediate\certs\intermediate.cert.pem > intermediate\certs\ca-chain.pem

C:\ca>type cacert.pem >> intermediate\certs\ca-chain.pem

C:\ca>c:\users\administrator\Cygwin\home\administrator\CW\_Tools\tools\openssl\OpenSSL-Win32\_0\_9\_8k\bin\openssl.exe verify -CAfile intermediate/certs/ca-chain.pem intermediate/certs/www.example.com.cert.pem intermediate/certs/www.example.com.cert.pem: OK  
  
**NOTE**: Notice the ‘trick’ of bundling the root CA and intermediate CA certs together for the test.  
  
*JeremyC 20/2/2016.*  
**END**