Updated:March 22, 2024 / By steve

Using A Lets Encrypt Certificate on Mosquitto

If you are running MQTT on a closed network then creating and using your own certificates as explained in Creating and Using Client Certificates with MQTT and Mosquitto is perfectly fine.

However if you require public access to the broker over SSL and in particular over websockets and SSL then using a public certificate like let's encrypt has advantages.

The main advantage of Public certificates have over self signed certificates is that they are natively supported by web browsers which is very important when using the JavaScript websockets client.

Let's encrypt certificates are popular because they are free and trusted (incorporated in popular browsers).

The main draw back of using them is that you need to go through the process of obtaining one which isn't as straightforward as when you purchase one.

Using Certificates with Mosquitto

To use certificates on Mosquitto you will need to

- Obtain (create) the certificates
- Install the certificates on the broker
- Configure clients to use them.

Creating Let's Encrypt Certificates

To get a certificate you need to have a valid Internet domain name and also:

- A web server running and access to the root folder of the web server

- Alternatively access to the DNS records of the domain name.

This is because as part of the verification process you need to add entries on the web server or alternatively add new DNS entries.

I will cover both methods in this tutorial but would point out that I didn't get the web server method to work, and also this method cannot be used for wild card domain names.

Therefore my preferred method is DNS.

The certificate management is done using the certbot command line tool which you will need to install.

Certbot integrates with many web servers and can be used to automate certificate requests and renewals.

However as I am going to be using the certificate on mosquitto and so I will be covering the manual method only.

When using the manual method you will also need to manually renew the certificate.

Certificates are valid for 2 years

Installing Certbot

To install certbot on Ubuntu use:

```
sudo apt-get -y install certbot
```

There are more detailed install instructions here if needed

Certificate Types

With Let's encrypt we can create a certificate for single host or for a wildcard domain a single host would look like this

mqtt.example.com

and a wildcard like this

*.example.com

The advantage of the wild card domain is that we could host a web server at:

www.example.com

and a mqtt broker at

mqtt.example.com

all using the same certificate.

However the web server and mqtt broker could share the same certificate without requiring a wild card domain as they use different ports and so are distinguishable from each other.

The only requirement is that you access them using this domain name.

General Process

- Run certbot with desired options.This can be done on any computer.
- Answer the prompts and stop when you get the instructions for making either the DNS changes or the web server changes.
- Go to your domain/web host and add files to the web server or text records to the DNS records.
- Complete process by pressing continue which should tell you that it worked or not.

Getting Certificates Examples

I will cover

- Getting a domain certificate using the web server method.
- Getting a wild card certificate using the web server method.

Getting a Domain certificate Web Server Method

In this method you request a certificate for a domain name and are then instructed to load file into a folder on the web server hosting that domain name.

Web servers can be accessed using the domain name or alternatively by convention using the domain name with a www prefix e.g

example.com or www.example.com

The command is

```
certbot certonly --manual --preferred-challenge http -d
```

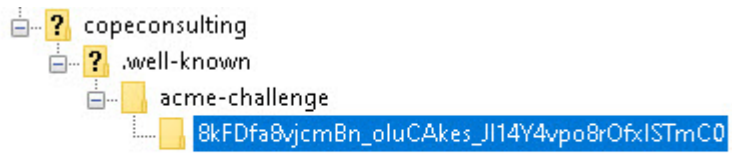
You can see that you are asked to create a file with a given name and place it in a folder on the website.

Once you have done this you click continue and the cerbot will go through the verification process and create the certificates if successful.

```
steve@mint2:~$ sudo certbot certonly --manual --preferred-challenge http -d copeconsulting.co.uk
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator manual, Installer None
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for copeconsulting.co.uk

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
NOTE: The IP of this machine will be publicly logged as having requested this
certificate. If you're running certbot in manual mode on a machine that is not
your server, please ensure you're okay with that.

Are you OK with your IP being logged?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Create a file containing just this data:

8kFDfa8vjcmBn_oIuCAkes_Jl14Y4vpo8rOfxlSTmC0.HFKmZbxLx8-c2wqvopbMBo9QF8ImKRecIVOJg9bL7q0

And make it available on your web server at this URL:

http://copeconsulting.co.uk/.well-known/acme-challenge/8kFDfa8vjcmBn_oIuCAkes_Jl14Y4vpo8rOfxlSTmC0

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Press Enter to Continue
Waiting for verification...
Challenge failed for domain copeconsulting.co.uk
http-01 challenge for copeconsulting.co.uk
Cleaning up challenges
Some challenges have failed.
```


In my case it failed. It did give a reason but unfortunately I forgot to take a screen shot. I also tried several times with the same result.

This is what my server folder looked like



Getting a Wild Card Certificate DNS Method

In this method you request a certificate for a wild card domain name and are then instructed to create a text record on your DNS server.

```
certbot certonly --manual --preferred-challenge dns -d *
```

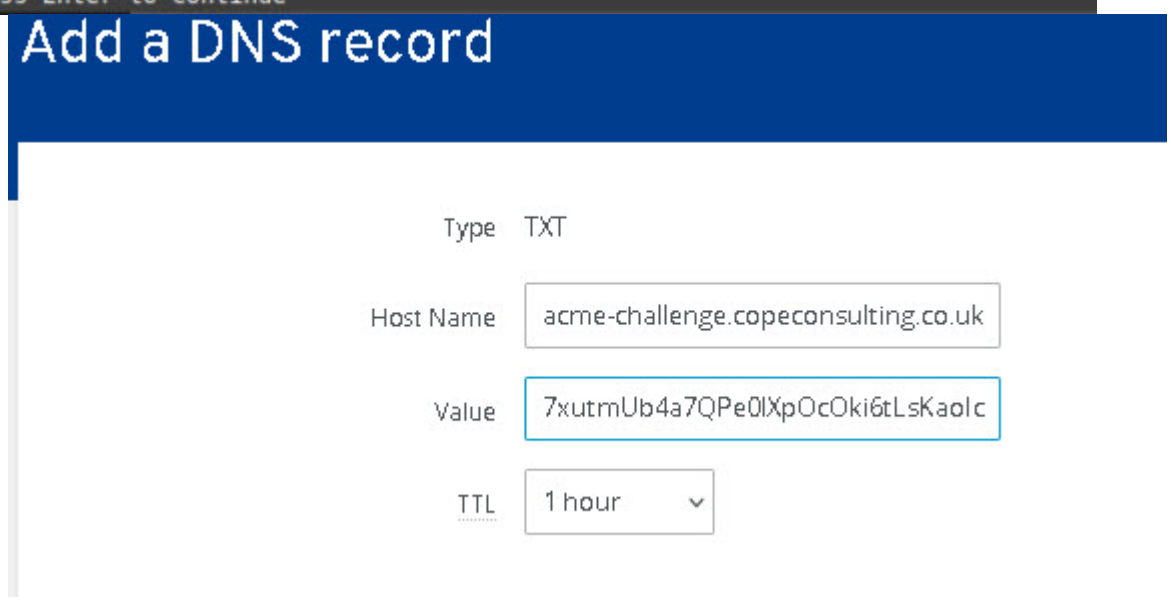
Notice the challenge is DNS and the domain name using the wild card character *.

```
steve@mint2:~$ sudo certbot certonly --manual --preferred-challenge dns -d *.cop
econsulting.co.uk
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator manual, Installer None
Obtaining a new certificate
Performing the following challenges:
dns-01 challenge for copeconsulting.co.uk

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
NOTE: The IP of this machine will be publicly logged as having requested this
certificate. If you're running certbot in manual mode on a machine that is not
your server, please ensure you're okay with that.

Are you OK with your IP being logged?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Note: You need to answer yes to IP being logged.

You now need to add text records to DNS on your Domain Name provider before continuing.

```
steve@mint2: ~
File  Edit  View  Search  Terminal  Help
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator manual, Installer None
Obtaining a new certificate
Performing the following challenges:
dns-01 challenge for copeconsulting.co.uk

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
NOTE: The IP of this machine will be publicly logged as having requested this
certificate. If you're running certbot in manual mode on a machine that is not
your server, please ensure you're okay with that.

Are you OK with your IP being logged?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please deploy a DNS TXT record under the name
_acme-challenge.copeconsulting.co.uk with the following value:

AvmGDZdEQ_M57xutmUb4a7QPe0lXpOcOki6tLsKaoIc

Before continuing, verify the record is deployed.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Press Enter to Continue
```

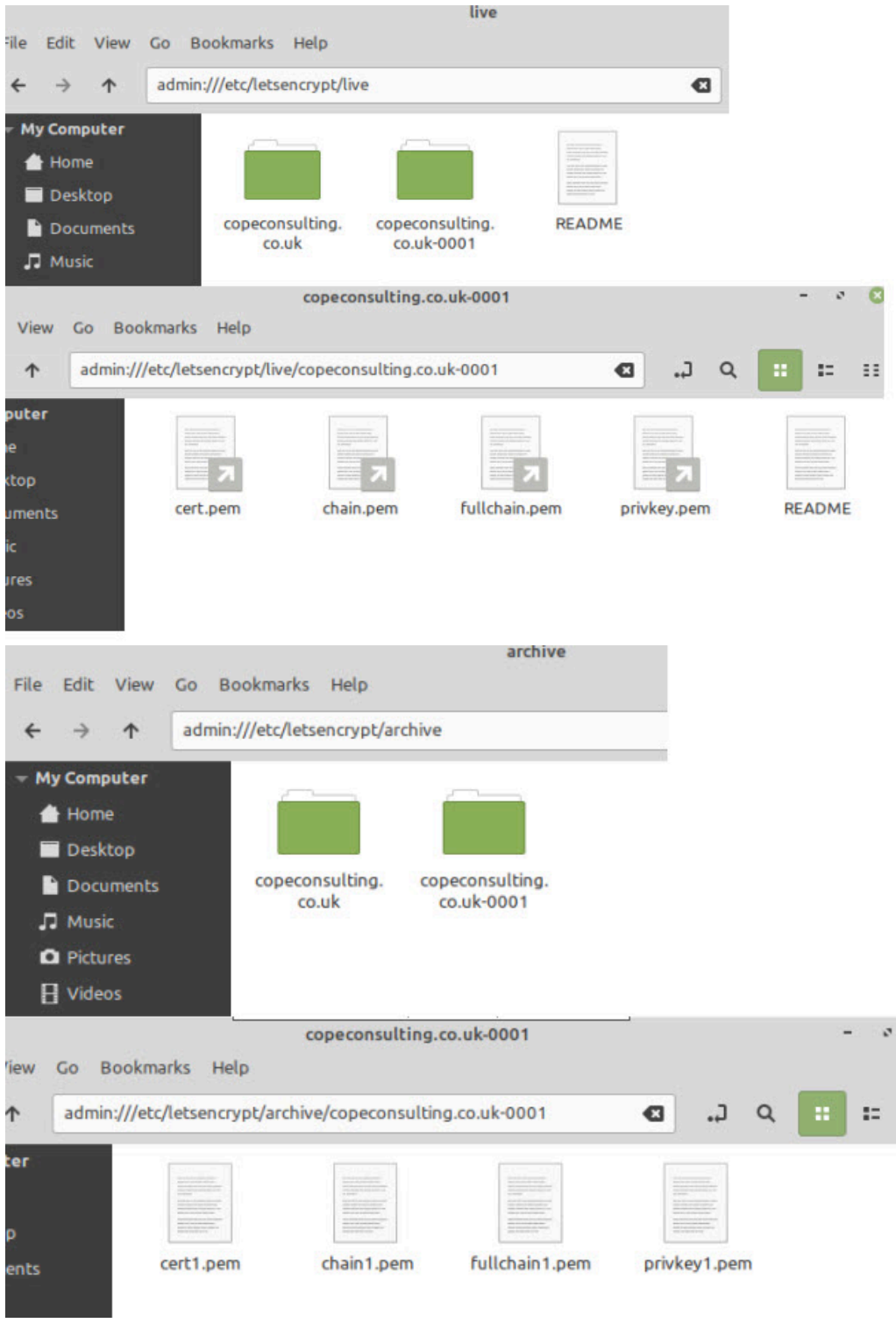


Now you click continue and you should get a verification successful message.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Press Enter to Continue
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/copeconsulting.co.uk-0001/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/copeconsulting.co.uk-0001/privkey.pem
   Your cert will expire on 2023-03-06. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot
   again. To non-interactively renew *all* of your certificates, run
   "certbot renew"
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le

steve@mint2:~$
```

Let’encrypt store all files under the folder /etc/letsencrypt/live/domainname/ folder.

If you look in this folder you will see symbolic links to the files which are actually stored in the /etc/letsencrypt/archive folder.



Using The Certificate Files on Mosquitto

On a running serving mosquitto expects the certificate files and keys to be in the /etc/mosquitto/certs folder.

You can either copy these files there or create symbolic links to the files in the archive folder.

Let's encrypt recommends the symbolic link method because the certificate renewal will work without the need for copying files.

However my attempt to do this failed and so I copied over the physical files from the archive folder into the /etc/mosquitto/certs/domain_name_folder/.

I also had to change the permissions on the privkey.pem file so that anyone could read it using.

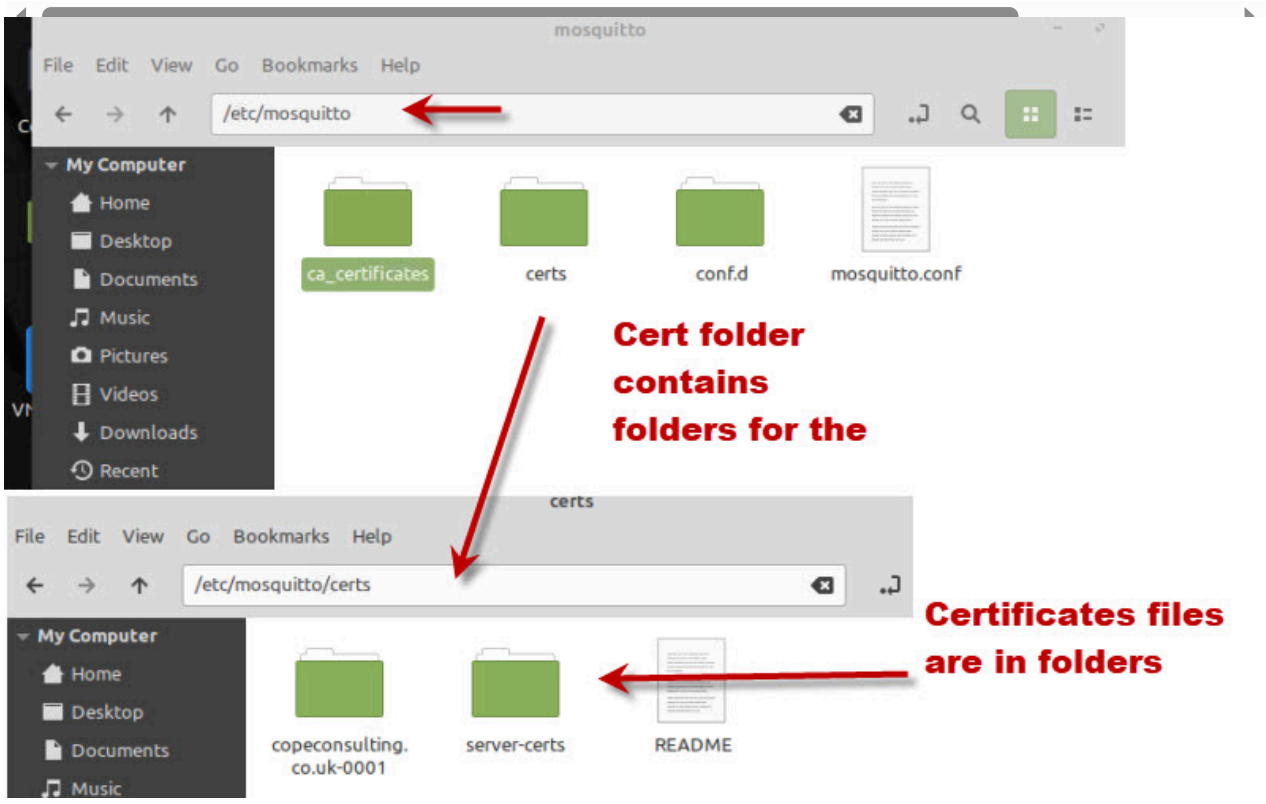
```
sudo chmod 755 privkey.pem
```

The entries in the conf file are show below:

```
listener 8883
cafile /etc/mosquitto/certs/copeconsulting.co.uk-0001/ch
```

```
keyfile /etc/mosquitto/certs/copeconsulting.co.uk-0001/p
certfile /etc/mosquitto/certs/copeconsulting.co.uk-0001/
```

This is what my /etc/mosquitto folder looks like:



Configuring Clients on Windows and Linux

On Linux trusted certificate files are stored in the /etc/ssl/certs folder.

This folder contains individual certificate files and also a bundled certificate file called **ca-certificates.crt**.

It shouldn't be necessary to add the ca to the trusted store as it is probably already there but the procedure is given below.

You can add new certificates (e.g new.crt) to this store using the following process on Ubuntu

- Copy ther CA to the /usr/local/share/ca-certificates/ folder using **sudo cp new.crt /usr/local/share/ca-certificates/new.crt**
- Update the CA store: **sudo update-ca-certificates**

If need be you can remove it by deleting the certificate file from the **/usr/local/share/ca-certificates/** folder and the running **sudo update-ca-certificates** again.

Reference here

Now you can either use the capath switch or cafile switch in mosquitto_sub/pub

–capath /etc/ssl/certs or use –**-cafile /etc/ssl/certs/ca-certificates.crt**

```
mosquitto_pub -h mqtt.copeconsulting.co.uk --cafile cert
```

On windows certificates are stored in the registry so are only usable by windows programs.

For utilities like mosquitto_pub/sub ,python or node clients then you need to download a copy of the ca-certificates.crt file and place it in a suitable location.

Download here

on my system I called the file **cacert.crt** and here is the an example using it

```
mosquitto_pub -h mqtt.copeconsulting.co.uk --cafile cert
```

Here is another example using the Python client on windows.

```
client.tls_set('c:/python34/steve/mqtt-demos/certs/cacer
```

Note: You need to use a version 3.6 or above

Testing Tips

I always test from my home folder and place the cert files and conf files in directories in my home folder which usually avoids any permission issues.

After testing I then move the files to the /etc/mosquitto folder and test from the command line using:

```
sudo -u mosquitto mosquitto -c /etc/mosquitto/mosquitto.
```

This runs mosquitto as the mosquitto user and you will see any problems on the console.

After that I simple start mosquitto as a service using

```
sudo service mosquitto start
```

Problems Encountered

As mentioned previously I didn't get the web server certificate request validation to work and I still don't know why.

With the actual files I encountered permission problems. From research on the Internet it seems setting the file permissions to **755** was the standard.

```
chmod -R 755 cert_folder
```

Testing Using a Public Broker

When configuring certificates it can sometimes be difficult to know if the cause of a failure is broker configuration or client configuration.

Fortunately there are public test brokers available configured with Let's encrypt certificates at test.mosquitto.org which you can use.

8081 : MQTT over WebSockets, encrypted, unauthenticated- Let's encrypt

8886 : MQTT, encrypted, unauthenticated-Let's encrypt

Common Questions and Answers

Q- I've created a certificate for example.com can I use it for a web server as well as mosquitto

A- Yes, but both must be accesible using example.com. Trying www.example.com will not work.

Q- I want to run my webserver on www.example.com and my mqtt broker on mqtt.example.com what should I do?

A- Create a wildcard certificate

Q- I'm using MQTT on an Intranet and am using MQTT with SSL. I will not be using websockets. Can I create my own certificate or do I need a public certificate

A- You can create your own and it is the best option. Let's encrypt requires a Internet accessible domain name. See Creating and Using Client Certificates with MQTT and Mosquitto

Related Tutorials and Resources

- SSL and SSL Certificates Explained For Beginners
- Mosquitto SSL Configuration -MQTT TLS Security
- Understanding and Using the Mosquitto Dynamic Security Plugin
- Creating and Using Client Certificates with MQTT and Mosquitto
- Configure a Mosquitto Bridge With SSL Encryption
- Quick Guide to The Mosquitto.conf File With Examples
- SSL or Payload Encryption Discussion Post

Please rate? And use Comments to let me know more

8 comments

Eli Spizzichino says:

January 17, 2024 at 3:16 pm

I think it's important to warn the readers, that this configuration with keys specified in the conf, should be used ONLY if mosquitto manage the handshaking directly, in case NGINX/Apache handles the websocket initial connection to proxy it to mosquitto NO KEYS should be used in the mosquitto.conf.
This took me two days to figure it out, hope it helps other

Reply

Mohammed says:

January 2, 2024 at 9:31 pm

Hello Steve,
Is there a way to use an encrypted connection without providing the –capath, i.e., similar to HTTPS. I followed your tutorial and installed the certificates from LetsEncrypt but I want my clients to connect using a secure (encrypted) connection without the need to provide them with key files. I want my clients to be able, for example, to publish a value using the following command:
mosquitto_pub -h mqtt.example.com -p 8883 -u username -P password -t /test -m 12
or using Python to connect like the following commands:
client.username_pw_set('username', 'password')
client.connect('mqtt.example.com', 8883)
Thank you,
Mohammed

Reply

steve says:

January 3, 2024 at 4:11 pm

Https is done by the browser and the CAs are already stored in the browser so that the user doesn't need to do anything.
With Python or the mosquitto_pub /sub clients you need to tell them where the ca is as there is no default location.
Rgds
Steve

Reply

Iain Sim says:

August 3, 2023 at 6:19 am

With letsencypt you place 3 pem files in your mosquitto conf file, then you go on to mention crt and certs you use cafile certs/cacert.crt. Is that a publicly available cert ? and is that what clients would use to connect?

Reply

steve says:

August 9, 2023 at 6:38 pm

Yes the link to download it is in the tutorial.
https://raw.githubusercontent.com/bagder/ca-bundle/master/ca-bundle.crt
also take a look at this
http://www.steves-internet-guide.com/configure-mosquitto-to-use-a-commercial-certificate-for-ssl/
Rgds
Steve

Reply

Dan Gould says:

March 21, 2023 at 8:22 am

The cause of the expired certificate error in MQTT Explorer is that the DST Root CA X3 cross-sign has expired. MQTT Explorer uses the electron library that has the expired cert in it. Until that is fixed, you need to import the ISRGROOTX1 LetsEncrypt root certificate in MQTT Explorer. Get that here:

https://letsencrypt.org/certs/isrgrootx1.pem.txt

Alternative is to force certbot to use the ISRG Root X1 CA, by adding

–preferred-chain “ISRG Root X1”

to the certbot command.

Reply

Andrew says:

March 11, 2023 at 11:59 am

The instruction to type “certbot certonly –manual –preferrred-challenge http -d copeconsulting.co.uk” has an extra ‘r’ in the word ‘preferred’, causing it to fail.

Reply

steve says:

March 11, 2023 at 1:24 pm

Hi

Sorry about that and tks for letting me know I've corrected it.

Rgds

Steve

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Post Comment

By Steve Cope