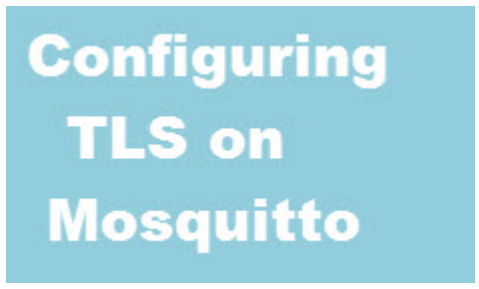
Updated:March 15, 2024 / By steve

Mosquitto MQTT Broker SSL Configuration Using Own Certificates



In this tutorial we will configure the mosquitto MQTT broker to use **TLS** security.

We will be using **openssl** to create our own Certificate authority (**CA**), Server keys and certificates.

We will also test the broker by using the Paho Python client to connect to the broker using a **SSL connection**.

You should have a basic understanding of **PKI**, certificates and keys before proceeding. See SSL and SSL Certificates Explained

The steps covered here will create an **encrypted connection** between the MQTT broker and the MQTT client just like the one between a web browser client and a Web Server.

In this case we only need a **trusted server certificate** on the Client.

We **do not need** to create client certificates and keys but this is covered in Creating and Using Client Certificates with MQTT and Mosquitto

Important Note: Many other tutorial on the web also configure **username and password authentication** at the same time. I don't recommend you do this as errors could be cause by either SSL or authentication. Only do one thing at one time when testing.

Client Requirements

- A CA (certificate authority) certificate of the CA that has signed the server certificate on the Mosquitto Broker.

Polls

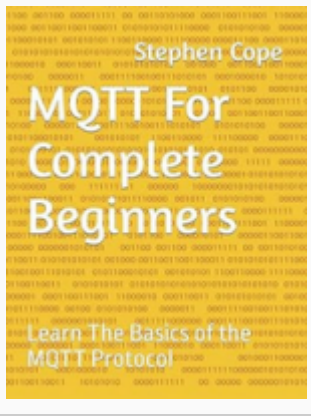
which client types do you use most often?

- C client
- Python Client
- Javascript Client
- Node-red Client
- Java Client
- Websockets client
- Arduino/ESP Client

Vote

View Results

Hi - I'm Steve and welcome to my website where you can learn how to build IOT systems using MQTT.



Broker Requirements

- CA certificate of the CA that has signed the server certificate on the Mosquitto Broker.
- CA certificated server certificate.
- Server Private key for decryption.

Creating and Installing Broker Certificates and keys

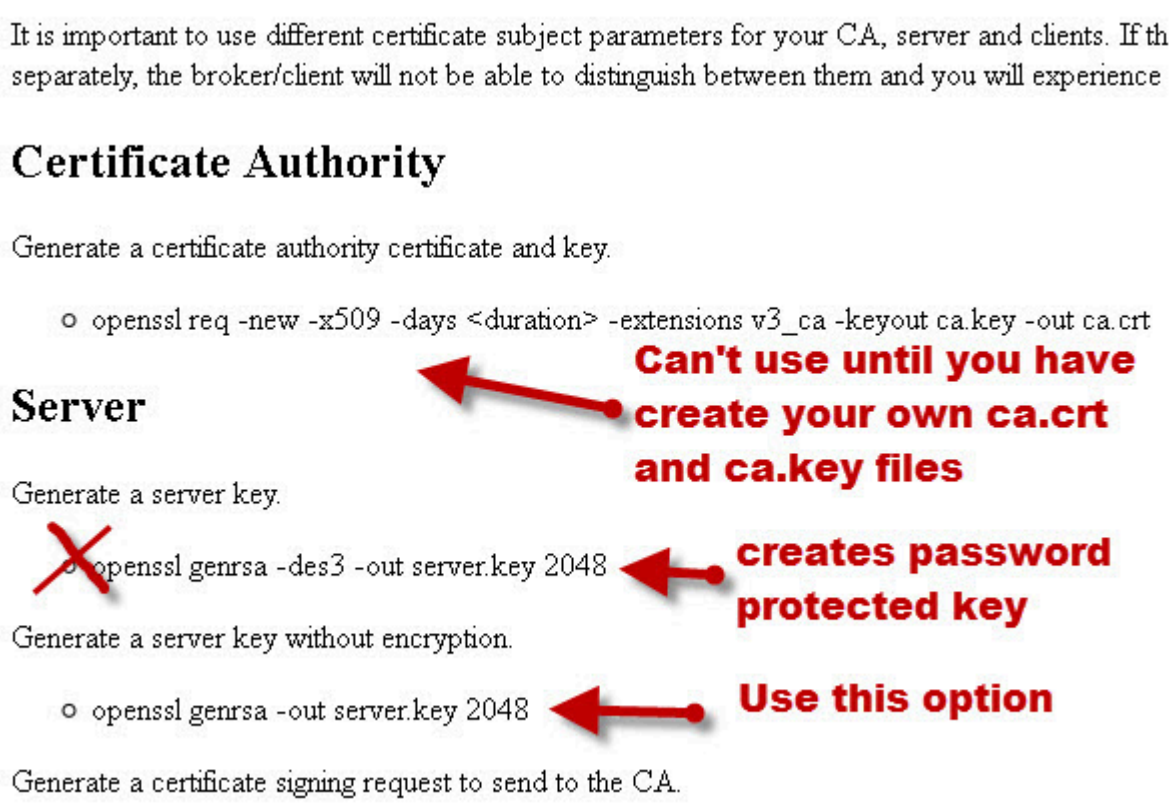
To create these certificates and keys we use the **openssl** software.

For windows you will find the install download files here.

On Linux you can install openssl using :

sudo apt-get install openssl

Although the commands to create the various certificates and keys are given in this Mosquitto manual page. Here is a quick snapshot:



There is a problem with the page because **openssl** no longer comes with a **CA certificate**, and so you will need to create your own **self signed CA certificate**.

You should also note that when you generate keys you **shouldn't use** encryption (the **-ds3** switch) for the server certificate as this creates a **password protected** key which the broker can't decode.

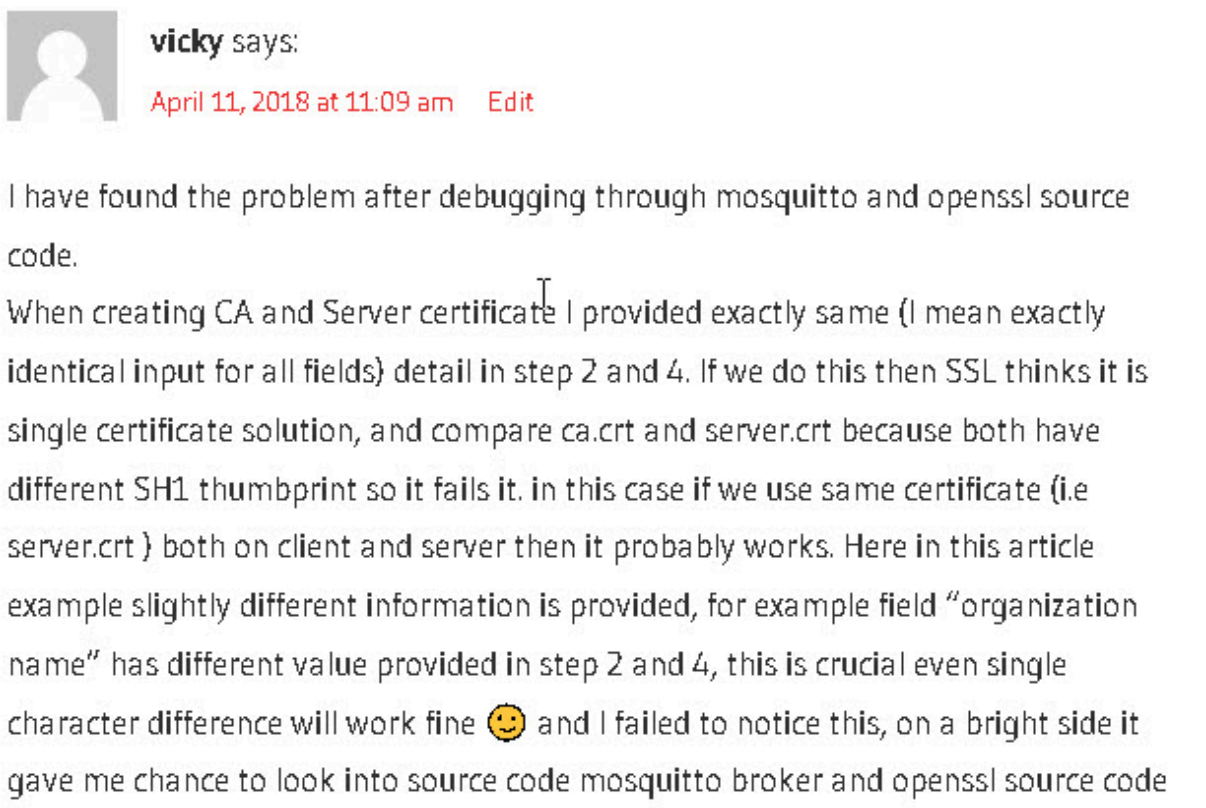
Note the certificates and keys created can be used on the Mosquitto broker/server, and also on a web server, which is why you see the term server used in the Mosquitto manual and not broker.

Overview of Steps

1. Create a **CA key pair**
2. Create **CA certificate** and use the **CA key** from step 1 to sign it.
3. Create a **broker key pair** don't password protect.

4. Create a **broker certificate** request using key from step 3
5. Use the **CA certificate** to sign the **broker certificate** request from step 4.
6. Now we should have a **CA key file**,a **CA certificate file**, a **broker key file**, and a broker **certificate file**.
7. Place all files in a directory on the broker e.g. certs
8. Copy the **CA certificate file** to the client.
9. Edit the **Mosquitto conf** file to use the files -details below
10. Edit the client script to use TLS and the CA certificate. -details below

Note: when entering the country, organisation etc in the form don't use exactly the same information for the CA and the server certificate as it causes problems. Here is a screen shot of a comment from a reader that brought it to my attention:



vicky says:

April 11, 2018 at 11:09 am Edit

I have found the problem after debugging through mosquitto and openssl source code.
When creating CA and Server certificate I provided exactly same (I mean exactly identical input for all fields) detail in step 2 and 4. If we do this then SSL thinks it is single certificate solution, and compare ca.crt and server.crt because both have different SH1 thumbprint so it fails it. in this case if we use same certificate (i.e server.crt) both on client and server then it probably works. Here in this article example slightly different information is provided, for example field "organization name" has different value provided in step 2 and 4, this is crucial even single character difference will work fine 🙂 and I failed to notice this, on a bright side it gave me chance to look into source code mosquitto broker and openssl source code

Detailed Steps

Note this as done on a windows XP machine.

The same commands and procedures apply to linux but the folder locations will be different and you may need to change permissions, as well as using the **sudo** command.

Step 1:

First create a key pair for the **CA**

Command is: **openssl genrsa -des3 -out ca.key 2048**

```
C:\steve>openssl genrsa -des3 -out ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................................+++
.....................+++
e is 65537 (0x10001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
```

Note: it is OK to create a password protected key for the CA.

Step 2:

Now Create a certificate for the CA using the **CA key** that we created in step 1

Command is: **openssl req -new -x509 -days 1826 -key ca.key -out ca.crt**

```
C:\steve>openssl req -new -x509 -days 1826 -key ca.key -out ca.crt
Enter pass phrase for ca.key:
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]:UK
State or Province Name (full name) [Some-State]:Shropshire
Locality Name (eg, city) []:Ironbridge
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CAmaster
Organizational Unit Name (eg, section) []:TEST
Common Name (e.g. server FQDN or YOUR name) []:ws4
Email Address []:steve@testemail.com
```

Step 3:

Now we create a server key pair that will be used by the broker

Command is: **openssl genrsa -out server.key 2048**

```
C:\steve>
C:\steve>openssl genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................................+++
.....................\.....................................................+++
e is 65537 (0x10001)
```

Step 4:

Now we create a certificate request **.csr.** When filling out the form the **common name** is important and is usually the **domain nam**e of the server.

Because I'm using Windows on a local network I used the Windows name for the computer that is running the Mosquitto broker which is **ws4**.

You could use the IP address or Full domain name. You must use the same name when configuring the client connection.

Command is: **openssl req -new -out server.csr -key server.key**

```
C:\steve>openssl req -new -out server.csr -key server.key
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]:UK
State or Province Name (full name) [Some-State]:Shropshire
Locality Name (eg, city) []:Ironbridge
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Server-cert
Organizational Unit Name (eg, section) []:test
Common Name (e.g. server FQDN or YOUR name) []:ws4
Email Address []:steve@testemail.com
```

Note: We don't send this to the CA as we are the CA

Step 5:

Now we use the **CA key** to verify and sign the server certificate. This creates the **server.crt** file

I**mportant Note Jan2023**– Due to problems with browsers requiring a SAN the command is now:

openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 360 -extfile filename

```
C:\steve>openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateseri
al -out server.crt -days 360
Signature ok
subject=/C=UK/ST=Shropshire/L=Ironbridge/O=MYCOMPANY/OU=MTDEPT\x1B[D\x1B[D\x1B[D
\x1B[D\x1B[D\x1B[3^\x1B[3^\x1B[C\x1B[C\x1B[C\x1BYDEPT/CN=wa4/emailAddress=steve@
testemail.com
Getting CA Private Key
Enter pass phrase for ca.key:
```

Note: scripts have also been updated see note at end

Step 6:

The above steps created various files. This is what the directory looks like now:

```
C:\steve>dir
 Volume in drive C has no label.
 Volume Serial Number is E091-351C

 Directory of C:\steve

06/11/2016  17:40    <DIR>          .
06/11/2016  17:40    <DIR>          ..
06/11/2016  17:35             1,419 ca.crt
06/11/2016  17:28             1,743 ca.key
06/11/2016  17:40                17 ca.srl
06/11/2016  17:40             1,359 server.crt
06/11/2016  17:39             1,143 server.csr
06/11/2016  17:37             1,675 server.key
               6 File(s)          7,356 bytes
               2 Dir(s)  19,440,615,424 bytes free
```

Need these 3 files

Note: We don't need to copy the CA.key file. This file is used when creating new server or client certificates.

Step 7:

Copy the files **ca.crt, server.crt** and **server.key** to a folder under the mosquitto folder. I have used a folder called **certs**.

on Linux you should already have a **ca_certificates** folder under **/etc/mosquitto/** and also a **certs** folder.

Use the **ca_certificates** folder for the **CA certificate** and the **certs** folder for the s**erver certificate and key**.

Step 8:

Copy the **CA certificate file ca.crt** to the client.

Step 9:

Edit the mosquitto.conf file as shown:

```
# Default listener
# ----------------------------------------------------------------
# IP address/hostname to bind the default listener to. If not
# given, the default listener will not be bound to a specific
# address and so will be accessible on all network interfaces.
# bind_address ip-address/host name
#bind_address

# Port to use for the default listener.
port 8883

# ----------------------------------------------------------------
# Certificate based SSL/TLS support
# ----------------------------------------------------------------
# The following options can be used to enable SSL/TLS support for
# this listener. Note that the recommended port for MQTT over TLS
# is 8883, but this must be set manually.
# See also the mosquitto-tls man page.
# At least one of cafile or capath must be defined. They both
# define methods of accessing the PEM encoded Certificate
# Authority certificates that have signed your server certificate
# and that you wish to trust.
# cafile defines the path to a file containing the CA certificates.
# capath defines a directory that will be searched for files
# containing the CA certificates. For capath to work correctly, the
# certificate files must have ".crt" as the file ending and must run
# "c_rehash <path to capath>" each time you add/remove a certificate.
#capath
cafile c:\mosquitto\certs\ca.crt
keyfile c:\mosquitto\certs\server.key
certfile c:\mosquitto\certs\server.crt
tls_version tlsv1
```

Edit default listener to use port 8883

Don't need capath So no need to do this c_rehash

Basic TLS Support on Mosquitto Broker

Notes:

1. I've used the **default listener** but you could also add an **extra listener**.
2. The **ca path** is not used as I told it the file location instead.
3. On my Linux install the entire **TLS section** of the **mosquitto.conf file** was missing I had to copy it from my windows install and then edit it. Here is the **mosquitto.conf file** documentation

Step 10 -Client Configuration:

Edit the client to tell it to use **TLS** and give it the path of the **CA certificate** file that you copied over.

I'm using the **python client** and the client method is tls_set(). Although there are several parameters that you can pass the only one you must give is the **CA file** as shown below.

client.tls_set('**c:/python34/steve/MQTT-demos/certs/ca.crt**').

The python client will default to TLSv1.

You shouldn't need to change it as the mosquitto broker also defaults to TLSv1.(before v 1.6)

However to change it to TLSv1.2 use:

client.tls_set('c:/python34/steve/MQTT-demos/certs/ca.crt',tls_version=2)

The **pub and subscribe scripts** that come with the mosquitto broker default to TLSv1.2.

Problems I Encountered and Notes

While creating and working through these procedures i encountered the following problems

1. Error when connecting due to the **common name** on the server certificate not matching.
2. I password protected the server key and the broker couldn't read it. I found this command which will remove the passphrase from the key – **openssl rsa -in server.key -out server-nopass.key.**
3. Not using the correct name for the broker. I used the IP address and not the name that I entered into the certificate.You can use the **tls_insecure_set(True)** option to override name checking as a temporary measure.
4. Authentication errors as I had previously configured my broker to require passwords. Therefore try to start with a clean conf file and beware that the errors you are getting may not be SSL related.

Self Signed Certificates

Currently the Paho python client require a CA certificate file and so it is not possible to use a self signed certificate. I came across a couple github threads relating to this but no real solution.

SAN Issues

I have had reports that the certificates do not work in a browser. see here. I'm not sure if this only applies to self signed certificates or also CA signed ones. The tutorial uses **CA signed certificates** and so do the scripts.

I have modified the scripts to require an external file. The DNS entries should match the common name you are using which is the DNS name or IP of the mosquitto broker.

The scripts have an example file included and you will need to edit it.

What is a SAN?

SAN (Subject Alternative Name) is an extension to the X.509 certificate standard that allows multiple domain names to be associated with a single SSL (Secure Sockets Layer) certificate. This allows a single certificate to be used for multiple website domains or subdomains.

Testing

If all goes well you should be able to publish and subscribe to topics as normal, but now the connection between client and broker is encrypted.

Unfortunately there is no easy way of seeing this.

This is the Python script I used:

```
import paho.mqtt.client as paho
import time
broker="ws4"
port=8883
conn_flag=False
def on_connect(client, userdata, flags, rc):
   global conn_flag
   conn_flag=True
   print("connected",conn_flag)
   conn_flag=True
def on_log(client, userdata, level, buf):
   print("buffer ",buf)
def on_disconnect(client, userdata, rc):
   print("client disconnected ok")
client1= paho.Client("control1")       #create client object
client1.on_log=on_log
client1.tls_set('c:/python34/steve/MQTT-demos/certs/ca.crt')
client1.on_connect = on_connect
client1.on_disconnect = on_disconnect
client1.connect(broker,port)           #establish connection
while not conn_flag:
   time.sleep(1)
   print("waiting",conn_flag)
   client1.loop()
time.sleep(3)
print("publishing")
client1.publish("house/bulb1","The Quick brown fox jumps over the lazy dogs tail
time.sleep(2)
client1.loop()
time.sleep(2)
client1.disconnect()
```

Must match common name on server certificate

To test using the mosquitto_pub client use:

```
C:\Python34\steve\mos>mosquitto_pub -h ws4 -t house/bulb1 --cafile
certs/ca.crt -m "test message" -p 8883
```

path to CA file

Broker address ws4

Failure Example

This shows that the common name you enter on the certificate must match the name used by the client when it connects. If not it doesn't work.

Use of Incorrect CA name - Although the IP address of steve-laptop is 192.168.1.206 it fails.

```
C:\mos> mosquitto_pub -h 192.168.1.206  -t test/topic -p 8883 --cafil
e c:\mos\certs\ca.crt -m message
Error: A TLS error occurred.

C:\mos> mosquitto_pub -h steve-laptop  -t test/topic -p 8883 --cafile
 c:\mos\certs\ca.crt -m message

C: mos>
```

Works Ok

Video -Configuring SSL on the Mosquitto MQTT Broker

How to Configure SSL on the Mosquitto MQTT Broker

TLS Versions

Starting with v1.6 I the support for tlsv1.1 was removed . You need to add the line

```
tls_version tlsv1.2
```

to your configuration file and when testing set the version e.g.

```
C:\mos>mosquitto_pub -h 192.168.1.41 -p 8883 -t test -m
```

You can see the change log here -https://mosquitto.org/ChangeLog.txt

File Permissions

One of the most common problems on Linux are file permissions.

The /etc/certs folder belongs to root and root has all permissions but the group others need read permissions to read the files.

Use ls-l to view permissions. Files with correct permissions are shown below:

```
steve@mint2:~$ ls /etc/mosquitto/certs/server-certs -l
total 12
-rw-rw-r-- 1 root root 1371 Dec  7 18:22 ca.crt
-rw-rw-r-- 1 root root 1249 Dec  7 18:22 server.crt
-rw-rw-r-- 1 root root 1675 Dec  7 18:22 server.key
steve@mint2:~$
```

Reported Problems and Solutions

- Wrong/Old openssl version reported on Centos 7. Update openssl fixed it.
- Problems when using capath on mosquitto_pub tool. Use cafile instead - mosquitto_pub -h host.name -u username -P password -t test/topic -p 8883 –cafile ~/keys/ca.crt -m message
- Problems with Server name on certificate. Use the **tls_insecure_set(True)** on the python client or the **–insecure** switch in the mosquitto_pub tool.

Useful OpenSSL Commands

Verify that a server certificate is signed by a particular CA. Use the Ca.crt file and the server.crt file.

```
openssl verify -CAfile ca.crt server.crt
```

it should return

```
server.crt: OK
```

Shell Scripts

To save you typing I've created two Linux shell scripts that run the commands and create server and client certificates and keys as in this tutorial and the client certificate tutorial.

Mosquitto Configuration Tutorials

- SSL and SSL Certificates Explained For Beginners
- Installing The Mosquitto broker on Windows and Linux
- Quick Guide to The Mosquitto.conf File With Examples
- Configuring and Testing MQTT Topic Restrictions
- Mosquitto username and Password Authentication Configuration Guide
- Configuring Logging on Mosquitto
- Configure Mosquitto Bridge With SSL Encryption- Examples
- MQTT Security Mechanisms
- Using A Lets Encrypt Certificate on Mosquitto
- Using SSL or Payload Encryption -Discussion Post

Other Related Articles and Resources:

- MQTT for Beginners
- MQTT and Mosquitto WebSockets Working Notes
- Beginners guide to PKI
- Hive MQTT security essentials TLS
- Trust anchors

Please rate? And use Comments to let me know more

379 comments

fakhri says:

April 29, 2024 at 1:00 am

C:\Users\Segni\Desktop\test\certif2>mosquitto_pub -h test.mosquitto.org -p 8883 -t test -m test –cafile C:\Users\Segni\Desktop\test\certif2\ca.crt –tls-version tlsv1.2 -d

Client null sending CONNECT

OpenSSL Error[0]: error:0A000086:SSL routines::certificate verify failed

Error: protocol error

Reply

steve says:

April 30, 2024 at 1:04 pm

It looks like you are using the wrong certificate. Did you download it from the site?

Rgds

Steve

Reply

Klaus says:

April 1, 2025 at 8:30 pm

Hello Steve.

I was able to successfully secure TLS for my Mosquitto broker following your description. I installed it on Ubuntu 24.04LTS. However, I also had to change the owner and group for the certificate files to mosquitto.

Regards, Klaus

Reply

steve says:

April 3, 2025 at 6:41 pm

Klaus

I will need to add that to the tutorial. It is something I forgot about as I usually do my tests in my home folder and so don't get the permission issues.

Rgds

Steve

Reply

Melodie says:

March 13, 2024 at 2:21 pm

Hi Steve, amazing guide! I understand that this was mainly done on local computer, but I am currently trying to implement this on a mosquitto docker image, namely eclipse-mosquitto:1.6.15-openssl. I have ensured that the CN of my rootCA.crt and mqtt_broker.crt are unique from each other, the path to the locations of rootCA.cert, mqtt_broker.crt and mqtt_broker.key inside the docker container contain the actual file (that have been loaded into the container using volumes) with file permissions allowing for read for all users. My client also has its own key and cert that is signed by rootCA.crt
Inside my mosquitto.conf, i have enabled the following:
listener 1883
listener 8883
protocol mqtt

cafile /mosquitto/config/certs/rootCA.crt
certfile /mosquitto/config/certs/mqtt_broker.crt
keyfile /mosquitto/config/certs/mqtt_broker.key
tls_version tlsv1.2

log_type all
ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384

allow_anonymous true
require_certificate true
use_identity_as_username true

However, when I use an openssl s_client or mqttx desktop client to connect to the mosquitto mqtt broker inside the docker container, I receive the following error message:
mosquitto | 2024-03-13T13:28:23: New connection from 172.19.0.1 on port 8883.
mosquitto | 2024-03-13T13:28:23: OpenSSL Error[0]: error:0A000086:SSL routines::certificate verify failed
mosquitto | 2024-03-13T13:28:23: Socket error on client , disconnecting.

I have tried to find any online sources where others have faced the same issue, but to no avail. Would you have any experience with this problem?

More contextual information:
– The mqtt_broker (mosquitto container) and mqtt_client are all connected to the same wifi connection and can reach each other.
– All certs for the broker and client are correctly signed as confirmed with the return of the command openssl verify being OK.
– Surprisingly, the client can verify the mqtt_broker cert during the TLS handshake, but it seems that inside the mosquitto the cert that it receives from mqtt_client cannot be verified, thus causing the error. That means the TLS handshake is completed (at least on the client side), but not on the broker side. When I set require_certificate to be false (so that mqtt_broker does not need to verify mqtt_client's cert), the 'tls' connection works with the log message:
mosquitto | 2024-03-13T14:17:06: New connection from 172.19.0.1 on port 8883.

mosquitto | 2024-03-13T14:17:06: New client connected from 172.19.0.1 as mqttx_63ee3df0 (p2, c1, k60).

– The CN of the certs for the mqtt_broker and mqtt_client are both localhost, but their OU are their respective component names. The CN of the rootCA is the IP address 192.168.255.121
– We have also included subject alternative names inside the certificate generation, where I included the IP address as well (eg IP.1 = 192.168.255.121) because I'm not sure if the CN not being an actual domain name (since we are using IP addresses) will cause the TLS to fail.

Thank you so much in advance for helping me with this! Greatly appreciate your other mqtt guides too ☺
Reply

steve says:

March 13, 2024 at 3:28 pm

The error look like a common name mismatch. The client should connect to the mqtt broker using the same ip address or domain name as configured on the certificate.
I have tried it on a docker image but if you contact me using the ask steve page and send me your cert I can try it.
Rgds
Steve

Reply

JCW says:

February 17, 2024 at 3:36 am

Is there a good reason to generate a new set of self-signed certificates for MQTT as opposed to using my existing certs for SSL on Apache? These are renewed as necessary by LetsEncrypt, and having to deal with another set of certs is just one more thing…

The mosquitto server is only for the local network. The only reason I need TLS is because I have a device that will only connect to the MQTT server over TLS.

Reply

steve says:

February 17, 2024 at 4:22 pm

No you should be able to use the existing ones.
rgds
steve

Reply

Arun says:

January 17, 2024 at 6:26 pm

Thanks Steve for the work.
Just a note for clarification: The “allow_anonymous” has to be set to true in mosquitto.conf file. This parameter is required to be set false when user name is required for client to connect. Here only ca certificate is used by client without user name so it has to be true. If allow_anonymous is set as false then a user name and password file has to be used.

Reply

steve says:

January 18, 2024 at 3:57 pm

Tks for that I will add it as a note.
rgds
steve

Reply

ghost says:

November 15, 2023 at 10:59 pm

Hi Steve,

Thanks for the tutorial!
I was wondering, you showed the screenshot of the /etc/mosquitto/certs/ directory including the keyfile with “correct” permissions. But I believe those permissions are incorrect because everyone on the system has access to the private key “server.key” or am I missing something? Since the “others” bit is set to read.

Wouldn’t it be better to have the mosquitto group as the group, and chmod 640 the private key?

Cheers

Reply

steve says:

November 16, 2023 at 6:42 pm

Hi
Not sure but will test. Because I do a lot of configuration on my system as it is used for creating tutorials I may have changed them.
However I think that they are the default permissions and the broker is stated by the mosquitto user which needs read permissions to the file so read on others would make sense.
Rgds
Steve

Reply

ghost says:

November 16, 2023 at 8:41 pm

Actually I would argue that it won't make sense to use the other bit. As the principle of least privilege is the best right? So by setting the readbit to others every other user can access the private key.

For example see this serverfault reference:

https://serverfault.com/questions/259302/best-location-to-keep-ssl-certificates-and-private-keys-on-ubuntu-servers#:~:text=To%20add%20to%20the%20private%20key%20location%3B%20make%20sure%20you%20secure%20it%20properly%20as%20well%20as%20having%20it%20in%20there.%20(chown%20root%3Assl%2Dcert%20and%20chmod%20640)

Thanks

Reply

steve says:

November 16, 2023 at 8:52 pm

You probably are correct but It is something I will check to see if it works with those permissions.
I also need to check the default permissions and owner on the folder on a clean install.
Rgds
Steve

Reply

Abdul Rasheed says:

November 15, 2023 at 8:49 am

I am using openssl 3.1.4 on windows 10 64bit.

Have an issue to generate server certificate in Step 5 above. First four steps are executed successfully and same information is given while generating step 5. I tried resolving issue using internet but not.
I know this is openssl issue, please help in resolving the issue.
Thanking in advance.
C:\Users\...>openssl x509 -req -in server.csr -CA ca.crt -CAKey ca.key -CAcreateserial -out server.crt -days 365 extfile server.key
x509: Use -help for summary.
more on this, on entering half command getting could not read CA private key, but when i double click and open am able to see all details.
C:\ssl\2023>openssl x509 -req -in server.csr -CA ca.crt

Certificate request self-signature ok

subject=C = SA, ST = Eastern, L = Alkhobar, O = Electronia, OU = R&D, CN = electronia.com, emailAddress = rasheeda@electronia.com

Could not read CA private key from ca.crt

Reply

steve says:

November 15, 2023 at 9:04 am

hi have you tried using the scripts

rgds

steve

Reply

Alex says:

November 13, 2023 at 12:53 pm

Hi Steve,

Thank you very much! That resource is really helpful!

"Currently the Paho python client require a CA certificate file and so it is not possible to use a self signed certificate. I came across a couple github threads relating to this but no real solution."

Can you please explain what you mean by this? "it is not possible to use a self signed certificate" for the Paho client or self signed certificate for the broker(server)? Does it mean Paho python client can't use TLS unless I purchase a certificate?

Many thanks.

Reply

steve says:

November 13, 2023 at 1:58 pm

Hi

No just create the certs using the scripts in the tutorial. It creates a new certificate authority and that works fine with Python.

Does that make sense.

Rgds

Steve

Reply

Givi Soltani says:

November 12, 2023 at 3:03 pm

Hi Steve,

Thanks to you and your book I did configure my "mosquito" broker and clients working

with SSL/TLS V1.2 on port 8883. However, This Rabbit Hole lead me to the “Trust Anchor” and I have been reading and trying to get it working, but there is hardly any organized steps/document/procedure to follow. Do you have/know of anything on that subject that can help me or start me on the right track? Thanks.
Regards,
Givi
Reply

steve says:
November 12, 2023 at 3:58 pm

The trust anchor term in actually new to me as well. I did look it up and it seems to generally mean the topmost certificate in the chain or the root certificate as I know it as.
I assume you are trying to configure SSL on AWS or something like that?
Rgds
Steve

Reply

Givi Soltani says:
November 12, 2023 at 8:17 pm

No, I prefer to stay in my own network. This setup with SSL/TLS V1.2 is working just fine, but I have come this far, why not continue. I came across the below links and information when I was searching for upgrading to SSL/TLS V-1.3 and Arduino sketches, such as MQTT example (with SSL!), when I read on these.

https://bearssl.org/index.html
https://github.com/OPEnSLab-OSU/SSLClient/blob/master/TrustAnchors.md
Thanks again, will be in touch.
Givi

Reply

JOHN PAUL DALA says:
September 20, 2023 at 10:38 am

Hi Steve,

Is it possible for the broker to use 2 or more CA certificates?

Reply

steve says:
September 20, 2023 at 1:49 pm

I would think it is but haven’t tried it.
Rgds

Steve

check and it seems not as there is a conflict with the server.crt that stops it loading.

rgds

steve

Reply

Jonathan says:

September 5, 2023 at 2:34 pm

Hi, thanks for the guide!

I have two identical IoT showers. one connects to aqualisa-mqtt.like.st and the other to aqualisa-mqtt-ringfence.like.st (Ive no idea why two identical showers have different servers, but it is what it is, Ive updated firmware, but both still insist on being different)

So I've configured my DNS to send them to my MQTT server IP and generated server certificates.

This has worked for one of the two showers, using the IP address of my broker in the certificates.

The other this has not worked for and I got this error

1693908652: New connection from 192.168.4.31:57420 on port 8883.
1693908653: OpenSSL Error[0]: error:0A000412:SSL routines::sslv3 alert bad certificate
1693908653: Client disconnected: Protocol error.

So I created new certificates using aqualisa-mqtt-ringfence.like.st in the certificates.

now I get this error instead

1693924374: New connection from 192.168.4.31:49605 on port 8883.
1693924374: OpenSSL Error[0]: error:0A000418:SSL routines::tlsv1 alert unknown ca
1693924374: Client disconnected: Protocol error.

The other shower continues to connect just fine and doesn't seem to care what the certificate has in it.

Any help would be much appreciated

Reply

steve says:

September 5, 2023 at 4:51 pm

When you say certificates do you mean certificate as both should use the same CA certificate. AS you are using your own local broker I would not bother with the certificate unless the shower needs to have one to work.
The unknown CA means that the CA you are using on the client isn't the one used on

the broker.

Rgds

Steve

Reply

Bernard says:

August 29, 2023 at 1:33 am

Error messages:

mosquitto.service – Mosquitto MQTT Broker

Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset: enabled)

Active: failed (Result: exit-code) since Tue 2023-08-29 09:24:30 +08; 16s ago

Docs: man:mosquitto.conf(5)

man:mosquitto(8)

Process: 2638 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exited, status=0/SUCCESS)

Process: 2639 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exited, status=0/SUCCESS)

Process: 2640 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited, status=0/SUCCESS)

Process: 2641 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited, status=0/SUCCESS)

Process: 2642 ExecStart=/usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf (code=exited, status=1/FAILURE)

Main PID: 2642 (code=exited, status=1/FAILURE)

CPU: 37ms

mosquitto.conf settings:

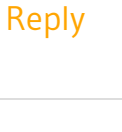
```
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

per_listener_settings true
pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d
allow_anonymous false
listener 1883
password_file /etc/mosquitto/passwd

listener 8883
cafile /etc/mosquitto/ca_certificates/ca.crt
certfile /etc/mosquitto/ca_certificates/server.crt
keyfile /etc/mosquitto/ca_certificates/server.key
```

tls_version tlsv1

When i comment out this line " #keyfile /etc/mosquitto/ca_certificates/server.key" broker able to restart, may i know what is the problem? FYI. I setting up mqtt broker server and flask backend running using the raspberry pi.

Another question is for the common name CA setup : What should i named with? my hostname static ip address is 169.254.199.134 and 10.10.01.225 is my broker ip address. Which one should i use?

Reply



steve says:

August 29, 2023 at 9:46 am

For the keyfile I would guess it can't find the file. On linux it is usually a permission issue. For this reason I always recommend testing with all files in the use home folder and starting mosquitto manually see http://www.steves-internet-guide.com/tips-for-testing-the-mosquitto-broker-on-linux/

the common name is the name you reach the broker so if you need to type

ping 10.10.01.225 to ping it then that is the common name.

If you have a local DNS server and can use something like

ping mybroker

then you can use mybroker as the common name.

Does that make sense?

Rgds

Steve

Reply

kaan says:

July 18, 2023 at 2:18 pm

broker = 'ws4' , client1.connect(broker, port)

when i run the code i get the error :

client1.connect(broker, port)

socket.gaierror: [Errno 11001] getaddrinfo failed

Reply

steve says:

July 18, 2023 at 2:37 pm

Hi

That is a DNS error try using the IP address.

Rgds

Steve

Reply

kaan says:

July 19, 2023 at 6:01 am

Thanks sir but when I try with my IP it gives error:
client1.connect(broker, port)
ssl.SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate (_ssl.c:1002)

Reply

steve says:

July 19, 2023 at 8:21 am

That is because you configured the certificate ti use the domain name so now you need to either configure a local DNS or edit the hosts file which is what I do. Take a look here if you are not familiar with the hosts file
http://www.steves-internet-guide.com/hosts-file/
rgds
steve

Reply

kaan says:

July 19, 2023 at 11:46 am

Thanks sir for answer, I already solve the problem with change the conf file settings. I don't it's the exact way but it works. Now my conf file looks like
listener 8883
protocol mqtt
allow_anonymous true

cafile C:\Program Files\mosquitto\certs\ca.crt
certfile C:\Program Files\mosquitto\certs\server.crt
keyfile C:\Program Files\mosquitto\certs\server.key
tls_version tlsv1.2
for your information...
thanks

steve says:

July 21, 2023 at 3:32 pm

Well done
Rgds
Steve

Bernard says:

May 23, 2023 at 3:42 am

Can I use it on raspberry pi 4?

Reply

steve says:

May 23, 2023 at 5:19 pm

I assume you mean mosquitto and SSL. If so then yes.

Rgds

Steve

Reply

Ron Lokey says:

April 26, 2023 at 1:31 pm

Steve, I ran you script to create server-certs, copied to folders, etc. and my client device connected perfectly. Then I changed the DNS in the v3.ext from DNS:mint2.home to my dDNS (xyz.690p.cz) and ran script again. Copied new certs to folders and restarted Mosquitto. Now my client does not connect? Do I need to replace Mint2.home with my dDNS name? Also, do I need to set up CronTab to renew there certs on a regualar basis?

Reply

steve says:

April 26, 2023 at 1:54 pm

Yes you need to replace mint2.home with your details. The scripts create certs with 10 year expiry so you should be OK. After that you just generate new ones and replace the old ones.

Rgds

Steve

Reply

Jessica Howe says:

April 11, 2023 at 2:27 pm

I am not sure that whether you are installing self-signed certificate or commercial, but I have some doubts from my side. I tried to read the article:

https://cheapsslweb.com/resources/self-signed-ssl-certificate-vs-trusted-ca-certificate

Can you just tell me that you are using self-signed or commercial CA certificate?

Reply

steve says:

April 11, 2023 at 2:37 pm

They are self signed certificates.

Rgds

Steve

Reply

Rossend says:

March 31, 2023 at 7:51 am

Followed all your steps but when restart mosquitto service it fails to start.

If I comment the line

#certfile C:\Program Files\mosquitto\certs\server.crt

mosquitto starts fine.

Reply

steve says:

March 31, 2023 at 9:45 am

Hi

It is usually because it can't find the file.Have you checked that it is in the right place and that you have permission to read it.

Rgds

Steve

Reply

Juergen says:

March 20, 2023 at 12:15 pm

additional Note: after generating the files several times manually I used your 2 sh-script files. I needed to modify the access rights and solved the problem with openssl verify (Ufff). But trying mosquitto_pub yields "Protocol error" and Mosquitto logs: tlsv1 alert inbternal error. Trying your Python script (with tls_version=2) I get ssl.certificaterError hostname doen't match either of 'mint2.home'….

SoO what does this mean how should the v3.ext-file that you deliverd with your scripts look like?

rgds

Juergen

Reply

steve says:

March 20, 2023 at 1:18 pm

Can you use the ask steve page to contact me and then you can email me your certs. The name mismatch is because you need to use the same name as you entered into the common name when you created the certificate.

mint.home the the name of the machine I tested it on and is on my network so that isn't the one to use.

Rgds

Steve

Reply

Steffen Mei says:

March 2, 2023 at 6:16 am

Dear Steve,

Thank you so much for your extensive guides! They help me really much. Since now we are after January 2023 and you updated the page with a hint towards SAN support for some browsers and how to handle this issue, I have some questions. The v3.ext file you provide in the example scripts describes the configuration of x509 extensions if I researched correctly. I understand the use of most of the config but have a question regarding especially "Subject Alternative Name". There you configure 3 DNS. Am I not understanding this configuration or may these be your local DNS? How shall I configure the v3.ext when I dont have a local DNS but instead am operating in the www (so maybe google dns?)?

I am looking forward to your response and wish you all the best!

Steffen

Reply

steve says:

March 2, 2023 at 3:39 pm

Hi

You don't need a DNS server. On my local network I configure

mint2.home

mint2.local

192,168.1.33

All of these I can use to access my MQTT broker. The only one that uses a local DNS server is mint2.home.

Does that make sense?

Rgds

Steve

Reply

Zafar says:

February 28, 2023 at 2:51 am

Hi Steve,

I recently started using MQTT and installed mosquitto 2.0, I generated the server and client certificate as the steps suggested by you in the above examples.

But while connecting I am getting the following error on the client side

mosquitto_sub -h 192.168.0.45 -p 8883 –cafile ./ca.crt –cert ./client.crt –key ./client.key -t sensors/drone01/altitude -d

OpenSSL Error[0]: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed

on the mosquitto broker side I am getting the following log

1677552453: OpenSSL Error[0]: error:0A000418:SSL routines::tlsv1 alert unknown ca

1677552453: Client disconnected: Protocol error.

Reply

steve says:

February 28, 2023 at 2:23 pm

This error is usually cause by using a different name on the server certificate than you use to access the broker.

I see that you use 192.168.0.45 to access the broker did you use this for the common name on the server certificate?

Id you add the –insecure switch an it works then it is a name issue as described

rgds

steve

Reply

Gary M says:

January 26, 2023 at 1:43 am

This is a great tutorial and was the primary source I used to get my Mosquitto broker setup with server and client connection authentication, etc. I ran into an issue thought that had me scratching my head for hours. I could not get the client to authenticate the connection to the server. Eventually I stumbled upon the issue where the server certificate was not recognized because the my server has both a localhost name and an IP address. I had to use SAN or subject alternative name when I created the server certificate in order to ensure that the both the IP and hostname where valid identification. This seemed to be the case eve when I used the IP when connecting from the client. I figured I would post this so that it may help someone else if they run into the same problem I did. Maybe this could even be added in the tutorial? Thanks again!

Reply

Anand says:

January 18, 2023 at 4:54 am

Hi steve,

I am getting below error ,

C:\Program Files\mosquitto>mosquitto_pub.exe -h localhost -t test/msg -m “hello” -p 1883 –cafile certs/ca.crt

Error: protocol error

My Conf file :

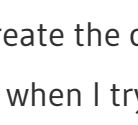
cafile C:\Program Files\mosquitto\certs\ca.crt

keyfile C:\Program Files\mosquitto\certs\server.key

certfile C:\Program Files\mosquitto\certs\server.crt

tls_version tlsv1

Reply



steve says:

January 18, 2023 at 11:17 am

Looks like you are using the wrong port

Rgds

Steve

Reply

Benjamin says:

January 16, 2023 at 7:32 pm

Hey Steve,

Thanks for this site, it is great to get me started with MQTT. Running into an issue with Certificates. Running on Windows 10 with a Linux VM on the same local machine. I was able to successfully Publish and Subscribe to a broker from both Windows and the Linux VM with a Windows based broker without any certs. I went through your steps to create the certs and everything matched what you had as far as I can tell. My problem is when I try to get the Certs to work. I made changes in the .conf file as needed as shown below:

port 7575

protocol mqtt

certfile C:\Program Files\mosquitto\certs\server.crt

tls_version tlsv1.2

keyfile C:\Program Files\mosquitto\certs\server.key

require_certificate true

cafile C:\Program Files\mosquitto\certs\ca.crt

allow_anonymous true

bind_address 0.0.0.0

When I try to subscribe or publish, I get the following error “OpenSSL Error[0]: error:14094418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca”

This is the command I am sending to subscribe that fails (WINDOWSPC = My PC Name) C:\Program Files\mosquitto>mosquitto_sub -t Message -h WINDOWSPC -p 7575 –cafile “C:/Program Files/mosquitto/certs/ca.crt” –tls-version tlsv1.2

I am using the same ca.crt file in both cases. From searching online, possibly this is the problem, but I am unclear and new to this. Any help is appreciated.

Reply

steve says:

January 16, 2023 at 8:09 pm

Haven't done this on windows for a long time. Try \ for the path in the mosquitto_sub command and see if it makes a difference.
rgds
steve

Reply

Dwarka says:

January 6, 2023 at 3:05 am

I run mosquitto broker on a Linux OS. I have followed all the steps, and when I restart my mosquitto broker, I get this error.
job for mosquitto.service failed because the control process exited with error code.
See "systemctl status mosquitto.service" and "journalctl -xe" for details.

Reply

steve says:

January 6, 2023 at 1:49 pm

Is this when you start mosquitto manually?
Rgds
Steve

Reply

Pretzelmeister says:

December 30, 2022 at 7:18 pm

Steven, good guide.
Two comments, please comment if I'm wrong:

1) You must a FQDN, if you run your server on the internet AND you want to validate the certficate. E.g. a IP address or 'simple' hostname (e.g. ws4) as Common Name does not suffice on the internet as the lookup fails.
This was not very clear above, yet as I understood it correctly that the common name must be a FQDN (not an IP address) as the lookup to verify the server identify will fail on a NON-FQDN name and on the IP Address as the Common Name. (in hind sight I see similar comments about this)

2) The current version of Mosquitto has change some of the config settings. Example "port 8883" is now "listener 8883". A neat error will appear outlining the required change.

Reply

steve says:

December 30, 2022 at 7:23 pm

Yes you must use a fqdn if you use on the Internet. On a local network a simple name likews4 or an ip address will work as long as you can ping it using that name/ip address.
Yes the use of port has been depreciated and you should use listener.
Rgds
Steve

Reply

Matthias says:

December 22, 2022 at 9:39 pm

Hi,

I run mosquitto on a Linux server and Owntracks on iPhone as client. This worked perfectly until recently when I changed my phone. Since then I can't get the connection back. I created new certificates but no lock. This is the error in the mosquitto.log:
022-12-22T22:37:00: OpenSSL Error[0]: error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown
2022-12-22T22:37:00: Client disconnected: Protocol error.

Any help would be appreciated.

Reply

steve says:

December 24, 2022 at 2:08 pm

Hi
It looks like you need to add the certificate to the trusted store.Access it using https in a web browser and although there isn't a page you get a certificate warning first and then you can click advanced and trust it. Then go back and try it.
Rgds
Steve

Reply

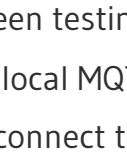
Matthias Reinagl says:

December 26, 2022 at 4:00 pm

Hi,
Thanks for your reply.
I tried that. First I got as you mentioned the certificate warning. But there was no option to install. In fact I do have the certificate installed on the phone. I compared the details from the one shown in the browser with the one installed. It is the same certificate.

I then also tried to click continue to the website. And now the question doesn't come back anymore (no surprise), so I can't check the same thing again.

Reply

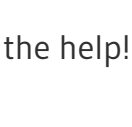


steve says:

December 26, 2022 at 5:17 pm

So after that it still doesn't work?

Reply



Matthias says:

December 28, 2022 at 2:08 pm

No.

Unfortunately not. Same error.

manoj says:

December 12, 2022 at 10:48 pm

Hi Steve

I have been testing a python client (the code is the same as yours with slight variations) with my local MQTT server. It is working fine with passwords and TLS as well. But when I tried to connect to my hiveMQ cloud, it fails to connect to the broker……..

Disconnecting reason 7 (Connection Return Codes 7)

MQTT Connection Return Codes 7 are only defined up to 5, can you please explain what causes this error? My cloud broker is;

CloudBroker2="f86f3a7384644b569094607fbe400e2e.s2.eu.hivemq.cloud"

Appreciate the help!!

Manoj

Reply

steve says:

December 13, 2022 at 11:51 am

Are you using mqtt v5 or 3

Rgds

Steve

Reply

joshua says:

December 9, 2022 at 9:11 am

Hi Steve,

Hope you are doing great.I got an error which says

"OpenSSL Error[0]: error:14094412:SSL routines:ssl3_read_bytes:sslv3 alert bad

certificate” in mosquitto. and for the same im getting errors ,

” esp-tls-mbedtls: mbedtls_ssl_handshake returned -0x2700

esp-tls-mbedtls: Failed to verify peer certificate!

esp-tls: Failed to open new connection

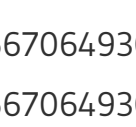
TRANSPORT_BASE: Failed to open a new connection

MQTT_CLIENT: Error transport connect”

in my Espressif idf. could you please help me out here?

Thanks in advance.

Reply



steve says:

December 9, 2022 at 11:25 am

Where is the error on the mosquitto console or the client?

Are you using client certificates?

rgds

steve

Reply

Will says:

October 29, 2022 at 9:39 am

Hi steve,

I have one question to setup bridge over TLS. Without certificate files, bridge can setup normally, but after add this configure, always say socket errors.

sh-4.4# /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf -v

1667064930: mosquitto version 1.5.1 starting

1667064930: Config loaded from /etc/mosquitto/mosquitto.conf.

1667064930: Opening ipv4 listen socket on port 8883.

1667064930: Opening ipv6 listen socket on port 8883.

1667064930: Warning: Mosquitto should not be run as root/administrator.

1667064930: Bridge local.auto2731evb-ivt-rtbm.test doing local SUBSCRIBE on topic #

1667064930: Connecting bridge test (192.168.168.38:8883)

1667064930: Bridge auto2731evb-ivt-rtbm.test sending CONNECT

1667064930: Socket error on client local.auto2731evb-ivt-rtbm.test, disconnecting

device side:

#capath

cafile ca.crt

keyfile client.key

certfile client.crt

PC side:

Both of certfile and keyfile must be defined to enable certificate based

TLS encryption.

cafile D:\Program Files\mosquitto\ca.crt

keyfile D:\Program Files\mosquitto\server.key

certfile D:\Program Files\mosquitto\server.crt

#tls_version tlsv1

Reply

steve says:

October 29, 2022 at 10:36 am

with a bridge the tls files need to go on the other broker as the bridge is a client so it needs configuring as a client.

If clients connect to the bridge then the bridge also needs cert files but as part of the standard configuration and not as part of the bridge.

Send me the config files of the bridge broker and the destination broker if you are still stuck.

Rgds

Steve

Reply

Mohammad Anas says:

October 3, 2022 at 2:37 pm

Hi Steve can you please explain me step no 8 from where I copy ca certificate file and where paste and secondly how to edit .config file it's not editable when I m going to edit config file its said you have no permission

Reply

steve says:

October 3, 2022 at 3:22 pm

You need to copy the ca to the client.The exact location will depend on the client you are using.

You need root permissions to edit the mosquitto.conf file.

for testing I use a conf file in my home folder and start mosquitto from the command line.

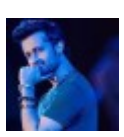
If you need to use the /etc/mosquitto/mosquitto.conf file then I copy it to my local folder and edit it then copy it back again.

Hoe that helps

Rgds

Steve

Reply



Muhammad Anas Uddin says:

October 3, 2022 at 8:51 pm

Hi thanks Steve can you explain me about client ? I didnt understand that you said client you used I have simple python script in which publisher publish 1 topic and subscriber subscribe the topic and I just want to test this script with ssl

features did you mean that I have to paste CA certificate in the folder where I saved my python file?

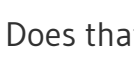
Reply

steve says:

October 4, 2022 at 12:13 pm

Yes. On the client it can be paced anywhere as long as the python client is configured with the fill path name.
Rgds
Steve

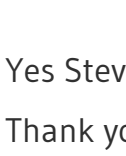
Reply



Sarath says:

October 11, 2022 at 4:04 am

Hey steve,
I have a question regarding the role ca.crt plays in the client. What does ca.crt mean to the client? Does it act as a public key?



steve says:

October 11, 2022 at 1:24 pm

The ca.crt is the certificate authority and is used to certify the server key. You actually use these all the time on the internet when you access a site that uses SSL i.e. https://
however you don't notice the ca.crt because the public ones are shipped in the browser.
However in the example we are creating our own certificate authority and so we need to copy of the certificate to any client that uses certificates signed by this CA.
Does that make sense?
rgds
Steve

Sarath says:

October 13, 2022 at 4:24 am

Yes Steve. Your explanation on October 11, 2022 at 1:24 pm was helpfull.
Thank you.
Regards,
Sarath

Sarath says:

September 28, 2022 at 4:27 am

Hello Steve,
I am working on enabling SSL in MQTT broker and clients. I was able to enable MQTT

broker with ca.crt,s erver.crt and server.key in mqtt broker(mosquitto in this case) and then i used the ca.crt in the client to communicate with the broker. This was successfully done. And then i was provided with a ca.crt alone(Self_Signed),i was not able to create a server key from this.
I am getting this error (Could not open file or uri for loading CA private key from ca.key) when signing the server crt with ca.crt(Provided by different team but still is a self signed certificate).
My questions are
1. Do i need the ca.key with me or should i send my server.crt and server.key to the team who created the ca.crt file
2. What all informations needs to be same in ca.crt and server.crt in the following fields
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
3. Can i create keystore and trustore for communication in java from the ca.crt or should i request the external team to generate them for me?

Thanks in advance.
Reply

steve says:
September 28, 2022 at 11:16 am

Hi
Certificates are the standard way. Client certs are for authentication not encryption. A better way I feel is to use payload encryption.
http://www.steves-internet-guide.com/encrypting-the-mqtt-payload-python-example/
https://stevesnoderedguide.com/encrypting-decrypting-mqtt-payloads
The mosquito bridge also supports shared key which I also prefer to certificates but the python client doesn't, not sure about the C client something I need to check.
Rgds
Steve

Reply

Sarath says:
September 28, 2022 at 11:36 am

Hi Steve, Thanks for reply. But the questions I asked here are different from the one I asked you through mail. Could you please questions 1,2 and 3.

Reply

steve says:

September 28, 2022 at 1:24 pm

Sorry My mistake
Don't understand exactly what you are doing this confused me
I am getting this error (Could not open file or uri for loading CA private key from ca.key) when signing the server crt with ca.crt(Provided by different team but still is a self signed certificate).

You have a ca certificate which works on the client. You use the same ca to create the client certificates in that is what you want to do.

If you have another team that is generating these for you then they need the Ca.crt and private key to do it.
Easier to send them all of the files your created when creating the server key.
Rgds
Steve

Reply

Sarath says:

September 29, 2022 at 4:05 am

You have a ca certificate which works on the client. You use the same ca to create the client certificates in that is what you want to do.
– No Steve. I am not creating a client certificate at client side. I am using the ca.crt to create the SSLSocketFactory and then use it with mqtt to conact the ssl mqtt broker.
– But on further reading, i cam to understand that in Java we only use keystore and trustore. So i tried to create keystore and trustore with the ca.crt we built in this blog.
– I am getting this error (Could not open file or uri for loading CA private key from ca.key) when i run the command – openssl pkcs12 -export -out eneCert.pkcs12 -in ca.crt
Ref:
https://docs.oracle.com/cd/E35976_01/server.740/es_admin/src/tadm_ssl_convert_pem_to_jks.html. Kindly help me in understanding how to convert our ca.crt or ca.key to keystore and trustore

steve says:

September 29, 2022 at 8:59 am

Hi
Sorry but I don't have any experience with oracle. I would have to do some research to find one and then I couldn't test it as I don't have the setup for it.
Maybe someone reading this has done it and can help.
Rgds
Steve

Nirav says:

August 15, 2022 at 7:04 pm

Hi Steve

I got everything working between MQTT broker (192.168.68.80) hosted as docker and MQTT.fx as a client. I use common name as raspberrypidev for both CA and Server which is my hostname. I can ping device with the hostname from another client. However when I try to use mosquitto_pub and mosquitto_sub on the broker itself I get "Unable to connect (Lookup error.)."

What could be wrong? Please advise.

Reply

steve says:

August 15, 2022 at 7:42 pm

Hi
Probable dns error. Have you check that you can ping the name? if not edit your hosts file on the client and add it.
http://www.steves-internet-guide.com/hosts-file/

Reply

A says:

August 15, 2022 at 12:26 pm

Hi Steve, thank you for the tutorial! I am creating my own mosquitto class from scratch using Paho.
May I please ask what the conditions are for the custom broker IP to work? I have tried creating TLS certificate with a domain name I own as I am the certificate authority, yet when I try to connect it via TLS, it doesn't connect.
For example:
client.tls_set(ca_certs=path_to_root_cert, certfile=cert_file, keyfile=key_file, cert_reqs=ssl.CERT_REQUIRED, tls_version=ssl.PROTOCOL_TLSv1_2, ciphers=None)

My mosquitto.conf:
cafile = mqtt/certs/ca.crt
keyfile = mqtt/certs/server.key
certfile = mqtt/certs/server.crt
tls_version tlsv1.2
require_certificate true

Did I do anything wrong here when setting TLS? Is it my certificate? I tried everything from localhost, to the domain name IP and still does not allow me to connect.

Reply

steve says:

August 15, 2022 at 12:53 pm

remove the require certificates true line

rgds

steve

Reply

Patrick Raatz says:

August 3, 2022 at 12:21 pm

Hello Steve,

I am generating the keys and certificates on an Ubuntu vm and using the mosquitto broker on an other windows computer in the same network.

For the CA and server-certificate I used the ip-adress of the windows computer as the domain name, and I dont know if I got that right.
When I try to verify the server certificate on ubuntu it fails.
When I try to publish a message on mosquitto with the ip-adress of the windows-computer I get the error: "A TLS error occured."
When I try to publish a message on mosquitto with the host-name of the windows-computer I get the error: "unable to connect (Lookup Error)."
I'm using MQTT-Explorer in Ubuntu with no luck so far obviously.

Could you help me with that?
Thanks in advance ☺

Reply

---

steve says:

August 3, 2022 at 12:58 pm

What is important is that the client needs to connect to the broker using the name or IP address. Unless you have configured the local hosts file or have a local dns server then using the name will not work so therefore using the IP address is the safest option.
This IP address you will need to use when creating the server certificate and it is the common name field.
Does that make sense?
Rgds
Steve

Reply

Andy says:

June 22, 2022 at 7:18 pm

Thank you very much for your effort to explain all that stuff about mosquitto and tls. Helped me a lot!

I'm stuck here with one last question:
I'd like to connect to my remote server using mosquitto_pub -h {ip-address} … However, my connection is refused due to the failed host name verification (using –insecure obviously circumvents the problem and it allows me to connect). Using my hostname for connection is no option here as it is a remote server. How can I still use host name verification and connect using mosquitto_pub -h {hostname}?

Thank you,
Andy

Reply

steve says:

June 22, 2022 at 7:26 pm

I'm assuming that the hostname isn't a valid dns host name as it would resolve. The only ways around this is to use the hosts table on your machine or a local dns server.
http://www.steves-internet-guide.com/hosts-file/
https://stevessmarthomeguide.com/home-network-dns-dnsmasq/
rgds
steve

Reply

María says:

June 15, 2022 at 4:55 pm

Hi Steve,

I would like to know if you could help me with the following problem because I have spent two days with this and I'm kind of stuck.

I have in a Rpi a mosquitto broker with a server TLS certificate signed by a self-signed CA located in the Rpi. I am trying to connect to this broker from a Parrot virtualbox machine using a python script with the following commands:

TLS_CERT_PATH = "/etc/mosquitto/ca.pem"
client_crt = "/etc/mosquitto/VM.pem"
client_key = "/etc/mosquitto/parrot.key"
client.tls_set(ca_certs=TLS_CERT_PATH, cert_reqs=ssl.CERT_REQUIRED,
tls_version=ssl.PROTOCOL_TLSv1_2, ciphers=None)
client.tls_insecure_set(False)
And the following error appears in the broker:

sslv3 alert bad certificate

And in the virtual machine:

certificate verify failed: IP address mismatch

I don't understand the error because if I run in the Virtual machine the following, where 192.168.1.254 is the IP of the Rpi:

mosquitto_pub -h 192.168.1.254 -p 2259 –tls-version tlsv1.2 –cafile /etc/mosquitto/ca.crt –cert /etc/mosquitto/VM.crt –key /etc/mosquitto/parrot.key -t Injecction_moulding/pressure -q 0 -m trying

It doesn't give me any error, even though I am using the same certificate files.

I thought that maybe it was something related to an intermediate certificate signing my Virtual machine's client certificate, but it is issued by the same CA that the broker uses. Moreover, I have also added in /etc/ssl/certs, the certificates that I am using just in case the CA was not recognising them as valid certificates.

Do you know what have I done wrong?

Reply

steve says:

June 15, 2022 at 6:37 pm

Hi
The
certificate verify failed: IP address mismatch
is usually because the common name on the server certificate is different than what you are using to access the broker.
This is easily fixed using
client.tls_insecure_set(True)

However I think that there is something else wrong and a common one is permissions for the cert files (Linux boxes).
Also you are using client keys which I don't recommend you do until you have SSL working correctly.
When working with cert files use you local home folder for testing and then move them to the /etc/mosquito/ folder one all is working.
S copy them to your home folder and double check the file permissions and then retry.
Let me know if you have any joy.
Rgds
Steve

Reply

María says:

June 15, 2022 at 7:25 pm

Thanks for your concern Steve! But it still gives me the same error although I am running the python script as sudo. The code related to the tls certificate is as follows:

```
client.tls_set(ca_certs=TLS_CERT_PATH, cert_reqs=ssl.CERT_REQUIRED,
tls_version=ssl.PROTOCOL_TLSv1_2)
client.tls_insecure_set(True)
publish.multiple(msgs =msgs, hostname = hostname, port=2259,client_id =
clientid, keepalive=60, will=None, auth=None, tls={'ca_certs': TLS_CERT_PATH },
protocol=mqtt.MQTTv311, transport="tcp")

hostname = "192.168.1.254"
TLS_CERT_PATH = "/etc/mosquitto/ca.pem"
client_crt = "/etc/mosquitto/VM.pem"
client_key = "/etc/mosquitto/parrot.key"
```

And it is not a problem of the rest of the code because I previously tried it without tls and it works.

Regards,

María

Reply

steve says:

June 18, 2022 at 7:56 pm

never used publish_multiple but tlsneeds to be a dictionary

dict = {'ca_certs':"", 'certfile':"", 'keyfile':"", 'tls_version':"", 'ciphers':"}

Rgds

Steve

Reply

Sergei Vlasov says:

June 8, 2022 at 2:18 pm

Hi Steve,

you stated: "Problems when using capath on mosquitto_pub tool."

Tried: true, there is a problem "a TLS error occurred". (I used *_pub version 2.0.10 from your pack)

Same error I see if I specify –tls-use-os-certs on the mosquitto_pub command line.

Could you please help guessing why these errors: code tracing shows that both options are related not to a Mosquitto code but to OpenSSL properties. (I want to use any of these options in my client)

My best guess was that the my CA certificate must be rehashed. If I use:

OpenSSL rehash C:\certs\

I'm getting an OpenSSL error message: " Not available; use c_rehash script".

Yes, I have the c_rehash.pl file in my OpenSSL pack. It is a Perl script … How to hook Perl to openssl – I failed to understand. Any hints?

Thank you very much. Great lessons.

Reply

steve says:

June 9, 2022 at 12:43 pm

Hi

Are you on windows? Have you tried by using the cafile rather than path.

Rgds

Steve

Reply

Sergei Vlasov says:

June 10, 2022 at 6:24 am

Yes, Win10. Yes CAfile works as described.

I generated self-signed certs using “localhost” for DN. Then, as expected I can connect with mosquitto (8883 port with TLS) from local machine only or from any machine by an IP address if “insecure” flag is set.

My efforts are intended to find a way for remote party to minimize certificates related troubles. (Anybody who learned TLS in-depth will understand me I think). Namely, I want, for example to easily connect with Mosquitto using the Distinct Name (network server name), of plain IPv4 address.

Rgds,

Sergei. (thank you for the reply)

Reply

steve says:

June 10, 2022 at 8:39 am

How do you want to connect. Is it via a web browser?

Windows store certificates in the registry. This might help https://community.spiceworks.com/how_to/1839-installing-self-signed-ca-certificate-in-windows

rgds

steve

Reply

María says:

June 6, 2022 at 11:49 am

Hi Steve,

Do you know how to send a certificate request from a Parrot OS virtual machine (located in a PC) to the CA located in the Rpi?

Thanks!

Reply

steve says:

June 6, 2022 at 11:59 am

It is a file so you can email it use a file share or put it on a usb stick.

Rgds

Steve

Reply

Maria says:

May 26, 2022 at 6:58 am

Hi Steve,

I followed the steps you explained but when I try to use a python client with client.tls_set it gives me the following error even though it can be seen that I have the ca.crt at the path that I have defined (it is c:/etc/mosquitto/ca_certificates/ca.crt):

context.load_verify_locations(ca_certs)

FileNotFoundError: [Errno 2] No such file or directory

Do you know where could I have done something wrong?

Reply

steve says:

May 26, 2022 at 8:14 am

Hi

Almost certainly a permission problem . When testing pace files in your local home folder so that you rule out any permission issues.

rgds

steve

Reply

Maria says:

May 26, 2022 at 9:27 am

Hi,

Thanks for your reply! But I moved the file into my home directory but it still gives me the same error with the command client.tls_set("c:/home/ca.crt").

Do you know what could be happening now?

Reply

steve says:

May 26, 2022 at 12:52 pm

Hi
it is still probably permissions either check them or create a new file and copy over the contents. I've had that problem before.
rgds
Steve

Reply

Maria says:
May 26, 2022 at 4:22 pm

Hi Steve,

I think is something related to the fact that I didn't use the same common name on the server certificate than the one of the CA. What could I do to fix it?

Sorry about asking a lot of questions but I am stuck on it.

Regards,
María

steve says:
May 26, 2022 at 4:47 pm

You need to use the same common name but that gives you an SSl error.The error you are currently seeing is a file not found error.
Rgds
Steve

Rebecca says:
April 21, 2023 at 9:42 am

Did you ever find a solution to this? I'm having the same issue

Reply

steve says:
April 21, 2023 at 11:47 am

What is the error exactly.
Rgds
Steve

Lukas says:
May 18, 2022 at 1:53 pm

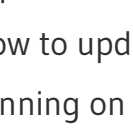
Hi Steve,

where is the key passphrase configured in Mosquitto for the encrypted keys?

Regards

Lukas

Reply



steve says:

May 20, 2022 at 8:05 am

Not sure what you mean here but the bridige can use psk see here

http://www.steves-internet-guide.com/mosquitto-bridge-encryption/

rgds

steve

Reply

Fabin says:

May 9, 2022 at 12:57 pm

Hi Steve,

Will an MQTT client can be configured to support both password based authentication and certificate based authentication at same time .The idea is to update the the client certificate using password based authentication if the certificate of the client gets expired .In all other cases we need a certificate based authentication.

How to update the certificate on the MQTT client side if it get expired . MQTT client is running on microcontroller based device which has support only for MQTT. Any other methods to do it. I am a beginner in this topic. Any information regarding this is appreciated.

Thank you in advance.

Reply

steve says:

May 9, 2022 at 3:28 pm

Hi

Don’ know it is something that I need to try and get back.

Rgds

Steve

Reply

Fabin says:

May 13, 2022 at 4:08 am

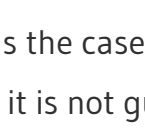
Hi,

Any information regarding my query.

Regards,

Fabin

Reply



steve says:

May 13, 2022 at 8:05 am

Sorry forgot will try to test it today

Rgds

Steve

Reply

steve says:

May 13, 2022 at 7:40 pm

Hi

It can be configured with both but I'm not sure if there is any precedence.

If you set a password file then you need to supply the a correct username/password regardless.

As far as I know there is not expiry checking on the broker but I need to test it to make sure.

If it is the case then they should be ok even when expired. However if it is the case now it is not guaranteed to remain so in the future.

You can overwrite the old cert by using a file transfer over MQTT.

I will amend the tutorial and drop you a note when I have tested further.

Rgds

Steve

Reply

Fabin says:

May 17, 2022 at 4:18 am

Thank you for the the information.

Reply

Andy says:

May 2, 2022 at 9:04 pm

Great tutorial but I noticed you are using the ca.crt on the client connection. Could I create individual client certificates and keys to make this more secure? I will be using MQTT with a lot of remote clients. Maybe the ca.crt is only used for the initial connection.

Reply

steve says:

May 3, 2022 at 12:55 pm

Hi

The single ca.crt is bacically the way secure Internet services like online banking work.

You can use client certificates instead of passwords see http://www.steves-internet-guide.com/creating-and-using-client-certificates-with-mqtt-and-mosquitto/

rgds

Steve

Reply

Azhaan says:

February 23, 2022 at 10:49 am

python3 mqtt_tls_pub.py #this is the filename
Traceback (most recent call last):
File "/home/amazhar/exp_mqtt/mqtt_tls_pub.py", line 21, in
client1.connect(broker,port)
File "/usr/local/lib/python3.9/dist-packages/paho/mqtt/client.py", line 914, in connect
return self.reconnect()
File "/usr/local/lib/python3.9/dist-packages/paho/mqtt/client.py", line 1044, in reconnect
sock = self._create_socket_connection()
File "/usr/local/lib/python3.9/dist-packages/paho/mqtt/client.py", line 3685, in _create_socket_connection
return socket.create_connection(addr, timeout=self._connect_timeout, source_address=source)
File "/usr/lib/python3.9/socket.py", line 844, in create_connection
raise err
File "/usr/lib/python3.9/socket.py", line 832, in create_connection
sock.connect(sa)
ConnectionRefusedError: [Errno 111] Connection refused

Reply

steve says:

February 23, 2022 at 3:55 pm

Hi

Not sure of the question but from the error message it looks like wrong IP or port.

Rgds

Steve

Reply

Azhaan says:

February 24, 2022 at 6:08 am

When I am executing the command using 1883 it's working fine and I am able to see the message in "mosquitto_sub"
The command that is working fine ->
$ mosquitto_pub -h 127.0.0.1 -t house/bulb1 -m "test message check!" -p 1883 –tls-version tlsv1.2

———

————-

But when I am changing the port to "8883" it is not working.
Error: Connection Refused.

———————————————————————————————

I am typing below some required information that may give you an idea of what I am doing.
Maybe you could find where I am doing the mistake.

———————

mqtt_tls_pub.py

———————————-

```
import paho.mqtt.client as paho
import time

broker="127.0.0.1"
port=8883
conn_flag=False
def on_connect(client, userdata, flags, rc):
global conn_flag
conn_flag=True
print("connected",conn_flag)
conn_flag=True
def on_log(client, userdata, level, buf):
print("buffer",buff)
def on_disconnect(client, userdata, rc):
print("client disconnected OK")
client1= paho.Client("control1") # create client object
client1.on_log=on_log
client1.tls_set("/etc/mosquitto/ca_certificates/ca.crt")
client1.on_connect = on_connect
client1.on_disconnect = on_disconnect
client1.connect(broker,port)
while not conn_flag:
time.sleep(1)
print("waiting",conn_flag)
client1.loop()
time.sleep(3)
print("publishing")
client1.publish("house/bulb1","The Quick brown fox jumps over the l>
time.sleep(2)
client1.loop()
time.sleep(2)
```

client1.disconnect()

---

mosquitto.conf

```
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

# MQTT over TLS/SSL
listener 8883
protocol mqtt
require_certificate false
#port 8883
#listener 8883
cafile /etc/mosquitto/ca_certificates/ca.crt
keyfile /etc/mosquitto/certs/server.key
certfile /etc/mosquitto/certs/server.crt

require_certificate false
tls_version tlsv1.2
allow_anonymous true
use_identity_as_username true

#listener 9001
#protocol websockets
```

While running mosquitto broker

****

$ mosquitto

```
1645682733: mosquitto version 2.0.14 starting
1645682733: Using default config.
1645682733: Starting in local only mode. Connections will only be possible from clients running on this machine.
1645682733: Create a configuration file which defines a listener to allow remote access.
1645682733: For more details see https://mosquitto.org/documentation/authentication-methods/
1645682733: Opening ipv4 listen socket on port 1883.
1645682733: Opening ipv6 listen socket on port 1883.
1645682733: mosquitto version 2.0.14 running
```

1645682733: New connection from 127.0.0.1:42548 on port 1883.
1645682733: New client connected from 127.0.0.1:42548 as auto-54CAFDE8-F785-2A23-D7AE-D2FFF1DA1B3A (p2, c1, k60).

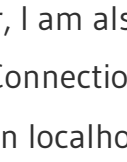
_____

above you can see that it is listening socket on port 1883

_____

I hope I have given the required details ….. Machine is UBUNTU 20.04

Reply



steve says:

February 24, 2022 at 9:44 am

hi
remove lines
require_certificate false
use_identity_as_username true

Reply

Srijan says:

February 15, 2022 at 3:01 pm

Hey Steve,

Thank you so much for the comprehensive tutorial. I followed the steps and have generated the CA cert and key, the server cert and key, and the client cert and key.
However, I am also getting this –
“Error: Connection refused” upon publishing.
It is all on localhost. Using a MAC, so Mosquitto Broker runs on 127.0.0.1 and port 1883.
My config looks like this –

cafile /Users/sdhare/MQTT/MosquittoServer/mqtt_ca.crt
certfile /Users/sdhare/MQTT/MosquittoServer/mqtt_server.crt
keyfile /Users/sdhare/MQTT/MosquittoServer/mqtt_server.key
require_certificate true
tls_version tlsv1.2

Works normally if I do “require_certificate false” in config. The certificates and keys are correct, I checked using SSL Verify: openssl verify -CAfile ca.crt server.crt

My Publish command looks like this –
mosquitto_pub -h 127.0.0.1 -t topic -m “Hello” –capath /Users/sdhare/MQTT/MosquittoClient/mqtt_ca.crt –cert /Users/sdhare/MQTT/MosquittoClient/mqtt_client.crt –key /Users/sdhare/MQTT/MosquittoClient/mqtt_client.key –tls-version tlsv1.2

What could be the possible reason?

Reply

steve says:

February 15, 2022 at 3:47 pm

Try adding

use_identity_as_username true

to the config file also try using the actual Ip address or the common name used in the certificate and not 127.0.0.1

Are you using the correct port as there is no -P option in the publish

rgds

steve

Reply

Srijan says:

February 16, 2022 at 8:22 am

Thank You Steve.

I think the common name is playing a role here. I am running both the Broker and Client locally. Earlier without TLS, both communicated by default on 127.0.0.1 and port 1883 (did not need to specify). I have a doubt here. After implementing TLS, would these two values change? Common name I gave my name of computer aka “apple” at time of certificate. I tried passing apple in place of 127.0.0.1 but it gives “Unable to connect (Lookup error.).” so I think 127.0.0.1 should be fine…

What is the significance of listener 8883? I have not used it. Should I?

Reply

steve says:

February 16, 2022 at 9:33 am

You need to add the apple entry to the hosts file see here

http://www.steves-internet-guide.com/hosts-file/

The 8883 port is the conventional port for SSL. Most use 1883 for MQTT and 883 for MQTT over SSL. However using 1883 for MQTT over SSL is ok but not conventional

Rgds

Steve

Reply

Srijan says:

February 21, 2022 at 9:40 am

Hi Steve,

Thank you for the reply.

1. What is the significance and difference between .conf keywords ‘Port’ and ‘Listener’?

2. It worked when I defined Port 1883 and Listener 8883 and did not pass any CA file when publishing from client: mosquitto_pub -h 127.0.0.1 -t topic -m “Hello” -p 1883
But I guess this method simply bypasses the TLS, right?

3. If I simply give port 8883 in .conf (listener commented out) then every pub command gives same error –
Client null sending CONNECT
OpenSSL Error[0]: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed
Error: A TLS error occurred.

4. I have checked the paths, they appear correct. Do the CA files necessarily have to be inside /etc/mosquitto/?

---

steve says:

February 21, 2022 at 3:05 pm

The use of port is discontinued in new versions but they are basically the same thing.
Yes you are bypassing ssl.
What clients are you using when you get the ssl serro are they javascript,C,Python,Java?
Rgds
Steve

Srijan says:

February 25, 2022 at 9:14 am

Thank you Steve. The problem is solved, it was the Common Name at time of generating the certificate. For the CA certificate I left it blank, and for Server certificate I entered the name of my PC/hostname for Common Name.
Then it worked fine. Earlier I had given my hostname in Common Name of both the certificates, but the Common Name should be different.

Next I want the Broker to accept the connections both with and without TLS. If a client wants to send certificate, then validate it, and if a client does not send certificate, that is also fine. How may I implement this?
Can both the clients connect to same port 8883 or the ports will be different, say 8883 for TLS and 1883 for non-TLS?

steve says:

February 25, 2022 at 9:18 am

Hi
Glad it is working. You need different ports for ssl+mqtt and mqtt
rgds
steve

Gerry Roston says:

December 14, 2021 at 7:56 pm

Stupid question alert (I think). I followed the instructions above, on my Raspberry Pi, and everything worked just fine. I am now trying to integrate this Mosquitto server with my Home Assistant system and am following the directions here: https://www.home-assistant.io/integrations/mqtt/. The various images don't match, but that is not my concern. However, under the heading 'Manual configuration', which shows how to modify the configuration.yaml file, I see lines for username and password. Based on the above instructions, what strings should I use for these??

Also, I assume that broker is the IP address of the MQTT server. If so, should that number be inside of double quotes?

Reply

steve says:

December 15, 2021 at 4:04 pm

Useranme and password can be left blank unless you have configured the broker to require it. The Ip address is probably in quotes.
Rgds
Steve

Reply

Alex says:

November 18, 2021 at 2:31 pm

Thanks a lot for you answer, it help me. Additional question, can I receive the tls-publishing message from the remote machine locally from the local mosquitto broker directly without using tls. Something like this:

hostA_192.168.1.10 $ mosquitto_pub -p 18883 -h 192.168.1.50 -t a/s/d/ -m "MSG=> Port:8883" -d –tls-version tlsv1.2 –insecure –cafile ca.crt –cert client.crt –key client.key

broker_192.168.1.50 $ mosquitto_sub -p 18883 -h 192.168.1.50 -t "#"

All examples I have found are when pub/sub is in the same situation, or both are locally or both are remote with a broker and both are using tls.
Really for subscribing I use paho.mqtt.c

or a local subscriber also should use tls (ca.crt, client1.crt, client1.key)?

Reply

steve says:

November 18, 2021 at 3:49 pm

A message published using TLS doesn't need to be received using TLS. for edge brokers local clients would probably publish without tls and TLS used to receive the data over the Internet,
Does that make sense?
rgds
Steve

Reply

Alex says:

November 8, 2021 at 4:44 pm

Hi, Steve!
Thanks for the great tutorial on mqtt and mosquitto. I've tried using your scripts to generate server and client certificates and keys. But after I created them, I verified them and the check failed.

_____

$ openssl verify -CAfile server-certs/ca.crt server-certs/server.crt
error 18 at 0 depth lookup: self signed certificate
error server-certs/server.crt: verification failed
$ openssl verify -CAfile server-certs/ca.crt client-certs/client.crt
error 18 at 0 depth lookup: self signed certificate
error client-certs/client.crt: verification failed

_____

What I have incorrect and how I can fixed it for create correct certificates and keys for my tests.
Thanks a lot Alex

Reply

steve says:

November 8, 2021 at 6:39 pm

Confirm that it gives a verify error and I will do some research on it.However the certificate will work I just check it.

Update

Have check it and the errors are probably because the certificate is not installed in the trusted store which it doesn't need to be for mosquitto as you specify the path. The self signed error is to be expected for the CA see https://security.stackexchange.com/questions/168564/what-is-the-difference-between-a-self-signed-root-certificate-and-a-root-certifi
Rgds
Steve

Reply

Brent says:

October 1, 2021 at 11:20 am

Hi I'm back again with another question. Is it possible for a client to connect over SSL to the mosquitto broker without passing in the ca.crt? My understanding is that upon connecting to a website, the server sends a message containing the server's SSL certificate and the client validates the certificate with its local trust store. Now, when I'm not my own CA and I generate the server's SSL certificate with certbot, would I still need to pass the ca.crt when connecting to my mosquitto broker since looking at my linux machine at /etc/ssl/certs I can see ISRG_Root_X1.pem is in here. This is the CA for Let's Encrypt.

Reply

steve says:

October 1, 2021 at 5:40 pm

The broker needs to be configured with the path to the ca,server key and and server certificate files. It doesn't make any difference that you are not the CA but you need to file.
Rgds
Steve

Reply

Brent says:

September 23, 2021 at 12:39 pm

Thank you for this amazing and well explained tutorial! However I have 1 question. If people want to connect to my mqtt broker they will need the ca.crt file. Is there an automatic way to distribute this file on connection like there is when connecting to a secure website?

Reply

steve says:

September 23, 2021 at 2:03 pm

When you connect to a secure website the ca.crt file is actually already installed in your browser.
So with mosquitto you will need to manually copy the crt file to the client machine.
The exception is if you use MQTT over websockets with SSL with a certificate from lets encrypt or another registered provider as this uses the certificate in the browser.
Does that make sense?
Rgds
Steve

Reply

Brent says:

September 24, 2021 at 7:27 am

That does make sense! Thank you so much for the quick response!

Reply

Jakob says:

July 14, 2021 at 5:13 pm

Hi Steve,

Thank you very much for some very useful tutorials. I would like to know a bit more about the pre-shared key-encryption setup. Does pre-shared key encryption mean, that only the payload, when the connection is created, is encrypted. Or does it also mean, that if you have psk-encryption setup, you will get TLS-encryption right from the get go, so no authentication data is transferred in clear text? So in that way psk-encryption can be used for a substitute to setting up the whole CA/client-thing.

Best regards,
Jakob

Reply

steve says:

July 14, 2021 at 5:40 pm

PSK is what you use on Wi-Fi. It is SSL but you choose the keys or passphrase as it is often called.
actually prefer it to the CA and certificates.
You can argue that it is not so secure because you are having to enter the keys manually and at each end of the connection but you do that on Wi-Fi.
Does that make sense?
rgds
steve

Reply

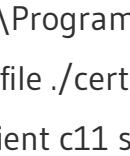
Maayan says:

May 2, 2021 at 8:31 am

Hi, Thanks for this information.
I am using the following SSL configuration:
a) client verify the server (default)
b) server verify client (required_certificate=true).
I supply for the server:
cafile, certfile and keyfile
I supply for the client:
bridge_cafile, bridge_certfile and bridge_keyfile
I would like to use with different CA certificates. Meaning, the server needs to know the

CA certificate of the client and the client needs to know the CA certificate of the server.
At this current configuration, I have to create the bridge_certfile with the same of CA certificate that has signed the server certificate
Is there configuration for that?

Reply



steve says:

May 2, 2021 at 8:42 am

It looks like a login problem. Mosquitto 2 doesn't allow anonymous access by default. Use the allow anonymouse true in the config file.
http://www.steves-internet-guide.com/mosquitto-broker/

Reply

Davide says:

May 1, 2021 at 10:41 pm

Thanks Steve for the great informations you provided in this article.
By the way after having followed your instructions to generate ca and server certificates, I started mosquitto broker and execute mosquitto_pub both on my machine and mosquitto_pub fails with CONNACK(5) error. Here is the log on client side

C:\Program Files\mosquitto>mosquitto_pub -h localhost -p 8883 -t /prova -m Ciao -d –cafile ./certs/ca.crt -i c11
Client c11 sending CONNECT
Client c11 received CONNACK (5)
Connection error: Connection Refused: not authorised.
Error: The connection was refused.

This is the log on broker side
1619908761: New connection from ::1:50581 on port 8883.
1619908761: Sending CONNACK to ::1 (0, 5)
1619908761: Client disconnected, not authorised.

Any idea to what could be the cause of this malfunction ?

Thanks

Reply

steve says:

May 2, 2021 at 8:43 am

It looks like a login problem. Mosquitto 2 doesn't allow anonymous access by default. Use the allow anonymouse true in the config file.
http://www.steves-internet-guide.com/mosquitto-broker/

Reply

Davide says:
May 2, 2021 at 10:15 am

Actually that was the cause. Thanks a lot

Reply

Michael Sandstrom says:
April 28, 2021 at 6:25 pm

Sorry, here is some more detail to my previous question.

This comes from Azure IoT.
– "There are two different ways to obtain a signing certificate. The first way, which is recommended for production systems, is to purchase a signing certificate from a root certificate authority (CA). This way chains security down to a trusted source.
The second way is to create your own X.509 certificates using a tool like OpenSSL. This approach is great for testing X.509 certificates but provides few guarantees around security. We recommend you only use this approach for testing unless you prepared to act as your own CA provider."

– So to break this down. For a real world scenario according to azure, we could purchase a CA signing certificate (be just as liable to guard this secret as if we were our own CA which Azure doesnt say here) and use this purchased certificate to sign CA certificates for devices?

Reply

steve says:
May 1, 2021 at 8:18 am

Exactly. I would go for own CA provided that 3 party access wasn't required and then you would need a public CA.
Rgds
Steve

Reply

Michael Sandstrom says:
April 28, 2021 at 6:06 pm

Hi Steve,
If I give x.509 certs a shorter lifespan I will have to have a PKI in place to be able to update these certificates securely. How do you recommend going about this process? I would either be using an IoT device with or without an OS. So placing the new certs could be done with SCP or over MQTT. Are there any services that offer this that you can recommend? I saw amazon has mqtt topics on the device that listen for requests to update certificates. https://docs.aws.amazon.com/iot/latest/developerguide/fleet-

provision-api.html. I am sure the topics are only accessible by admin users and are locked down sufficiently.

This aspect of PKI is definitely a critical part of any iot deployment. If we are using open source brokers and dont want to use amazons or azures iot brokers–what ways do you recommend (either a service or a diy solution) for implementing a PKI that can be effective at updating/managing client certificates. I'm assuming a seperate service/ database on the server would be needed that monitors the expiration dates, keeps track of the authenticated state of the devices, and performs certificate provisioning would be needed. Would creating intermediate CA certificates from the root and using this to sign the server & client certificates be better than signing by the head Root CA? The keys used to generate these would be stored offline. Then in the "ca_certificates" part of the mosquitto conf we would have a certificate with the whole chain of trust up to the root CA?

Thanks for any input. This is an interesting topic and is definitely important for the lifecylce of our devices.

Reply

steve says:

May 1, 2021 at 8:15 am

Interesting I will take a look at the provisioning guide

Reply

Susana says:

April 22, 2021 at 10:08 am

Hi Steve,

I tried to test the system and I'm having some problems regarding the sockets. This is the code that works:

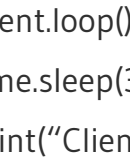
```
import paho.mqtt.client as paho #Import library
import time
import ssl

broker_address="192.168.1.44"#Broker IP
#broker_address="mqttserver" #Common name on server certificate
port=8883
conn_flag=False

username ="User1"
password = "test1"

def on_connect(client, userdata, flags, rc):
global conn_flag
conn_flag=True
print("connected",conn_flag)
conn_flag=True
```

```
def on_log(client, userdata, level, buf):
print("buffer",buf)
def on_disconnect(client, userdata, rc):
print("disconnected ok")

client = paho.Client("PythonClient")#Create an instance
print("Creates OK")
client.on_log=on_log
client.username_pw_set(username, password)
print("Username and passqword OK")
client.tls_set('/home/user/ca.crt',tls_version=2) #TIs version v1.2
client.tls_insecure_set(True) #To use the IP
print("TLS OK")
client.on_connect=on_connect
client.on_disconnect=on_disconnect
client.connect(broker_address,port)
while not conn_flag:
time.sleep(1)
print("waiting",conn_flag)
client.loop()
time.sleep(3)
print("Client publishing")
client.publish("office/room1/luminosity",111)
print("Data published")
time.sleep(2)
client.loop()
time.sleep(2)
client.disconnect()
```

Otherwise, when i change the tls.insecure_set to false [client.tls_insecure_set(False)] and i set the broker with the common name of the certificate [#broker_address="mqttserver"] i receive the following error: socket.gaierror: [Errno -2] Name or service not known.

Any idea what might be going on and how to fix it?

Reply



steve says:

April 22, 2021 at 3:51 pm

It looks like a name resolution problem,

use

ping mqttserver

if that doesn't work then the client can't resolve the name.To make it work you can add an entry to your local hosts file.

http://www.steves-internet-guide.com/hosts-file/

rgds

steve

Reply

Ramchandra Hegde says:

April 15, 2021 at 2:30 am

Hello Steve,

I love this article, it got me started on the topic. But I faced a few issues while deploying on client's premises. I have posted them on the stack overflow.

https://stackoverflow.com/questions/67074372/how-to-deploy-mqtt-with-ssl-on-internet-with-port-forwarding

I managed to solve the issues with following modification to step 5

openssl x509 -req -in server.csr -CA ca.crt -extfile v3.ext -CAkey ca.key -CAcreateserial -out server.crt -days 360

Where v3.ext contains

authorityKeyIdentifier=keyid,issuer

basicConstraints=CA:FALSE

keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment

subjectAltName=DNS:Enterprise, IP:111.6.77.122

It would be nice if you could add the section on SANs with more explanations.

Reply

steve says:

April 15, 2021 at 11:59 am

Tks for that I will try and take a look

Egds

Steve

Reply

Mathias Maier says:

December 28, 2022 at 12:06 am

I had the same problems: mosquitto with SSL worked fine. Mosquitto over websockets (javascript) worked fine. But mosquitto over websockets with SSL certificates got me certificate errors in the browsers. Reason: the certificates, when created like in your (otherwise excellent) guide, were missing the subjectAltName (SAN) that current browsers require because this is where they (have to) look for the domain name (additionally), not just in the DN. It took me days to figure this out. So for anyone who gets certificate errors with a websockets + SSL setup. Be sure to do what Ramchandra recommends. I didn't see his post but found out what to do by reading these two ressources: https://support.mozilla.org/en-US/questions/1379667
https://gist.github.com/KeithYeh/bb07cadd23645a6a62509b1ec8986bbc

Reply

steve says:

December 28, 2022 at 6:43 pm

Tks for the links I will try it and incorporate it into my shell scripts.

Rgds

Steve

Reply

Ramchandra Hegde says:

April 13, 2021 at 9:46 am

Hello Steve,

Can first five steps be added to batch file to automate the certificate generation. This includes human inputs being replaced by some other method?

Reply

steve says:

April 13, 2021 at 1:47 pm

Yes I have shell scripts that you can modify The only thing you might need to leave is the common name as that must be unique to the server.

rgds

steve

Reply

Mike Sandstrom says:

April 12, 2021 at 11:11 pm

if using tls over web sockets, how does the mqtt client retrieve the tls cert from the browser? I'm using a localhost webserver with mqttjs talking to a remote mosquitto broker over web socket with password authentication. Could I create certificate with letsencrypt, add it to the mosquitto.conf web sockets listener, and then how do I retrieve this certificate on the browser end? Thank you for any input, this has been spinning in my head for a while

Reply

steve says:

April 13, 2021 at 1:50 pm

Letsencrypt certificate should already be in the browser certificate store. You cannot use let's encrypt on a home network the server needs to be on the Internet.

Rgds

Steve

Reply

Ramchandra Hegde says:

April 9, 2021 at 6:14 am

Hello Steve,

I followed the steps and generated the files. I want to use these files with Eclipse Paho Java client, but there is no sample code that explains how to do this. Can you help?

Reply

steve says:

April 9, 2021 at 7:51 am

Take a look here
https://www.hivemq.com/blog/mqtt-client-library-enyclopedia-hivemq-mqtt-client/

Reply

Ramchandra Hegde says:

April 9, 2021 at 10:58 am

Hello Steve,
Thanks for the reply, my question got posted before I could add the details.
I am using following code sample to connect to MQTT from one year now. Of course I edited it to our needs. But our client wants more security, hence we need to implement SSL.
https://github.com/eclipse/paho.mqtt.java/blob/master/org.eclipse.paho.sample.mqttv3app/src/main/java/org/eclipse/paho/sample/mqttv3app/Sample.java.
It seems to be asking for JKS format. Can I use the keys generated here or I have to convert them and how?
Few hours back I posted the question on stack overflow also , but they closed it asking for more details.
https://stackoverflow.com/questions/67016117/how-to-connect-to-mqtt-broker-with-ssl-using-java-client

Reply

steve says:

April 9, 2021 at 3:09 pm

Yes you will need to convert them This may help
https://docs.oracle.com/cd/E35976_01/server.740/es_admin/src/tadm_ssl_convert_pem_to_jks.html
rgds
steve

Reply

Ramchandra Hegde says:

April 11, 2021 at 1:33 am

Yes, it works thanks.

Mike Sandstrom says:

March 15, 2021 at 5:15 pm

Hi Steve,

I've been receiving this error after following the tutorial

1615828489: Error: Unable to load server certificate "/etc/mosquitto/certs/server.crt". Check certfile.

1615828489: OpenSSL Error[0]: error:140AB18E:SSL routines:SSL_CTX_use_certificate:ca md too weak

Any idea how to resolve this?

Reply

steve says:

March 15, 2021 at 5:47 pm

Check the file permissions

Reply

Mike Sandstrom says:

March 15, 2021 at 6:20 pm

ok, they belong to my user on my machine which has sudo priviledges. I did chown mosquitto on the certs and ca_certicates file and still receive the same error.

I saw the same error on this stackoverflow post – https://stackoverflow.com/questions/52218876/how-to-fix-ssl-issue-ssl-ctx-use-certificate-ca-md-too-weak-on-python-zeep

I re-followed the steps above and added -shah256 to the command "openssl req -out server.csr -key server.key -new" this also has left the same error.

Reply

steve says:

March 15, 2021 at 7:25 pm

Move all the files to your home folder and run mosquitto from the command line that will tell you if there is a permission issue

Reply

Mike Sandstrom says:

March 15, 2021 at 8:29 pm

the certs are generated with sha1. Which I believe throws this error in some OS distributions. I think they need to be generated with something along the lines of sha256 I am using Ubuntu 20.10

Mike Sandstrom says:

March 16, 2021 at 2:45 pm

I've been receiving this error and have been troubleshooting in vein. I followed the steps above and for the client certificates. I added -sha256 when signing the CRT for the client and the server certificates which resolved the error saying the hash was too weak. I created the certs on my laptop which is in the Eastern Standard time and my server uses UTC time. Could this be the issue?

615905560: New connection from MYIPADDRESS on port 8883.
1615905560: Client disconnected due to protocol error.

---

steve says:

March 16, 2021 at 3:16 pm

Mike sue the ask steve page and contact me you can then email me all of your files and I will test them on my machine
http://www.steves-internet-guide.com/ask-steve/

Mike Sandstrom says:

March 15, 2021 at 8:39 pm

Solved! Sorry for the influx of responses. But the strength of server.crt was certainly the problem. I recreated it using sha 256 like so "openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server2.crt -sha256 -days 360" and added it to the certs file and mosquitto was able to start successfully! I will open an issue on mosquitto's github to let them to know to update their documentation. I am using Ubuntu 20.1

Reply

Mike Sandstrom says:

March 16, 2021 at 3:32 pm

This is what I see when running sudo lsof -i :8883. It doesnt look likes there is an "ESTABLISHED" connection.

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
mosquitto 1320 mosquitto 5u IPv4 40355 0t0 TCP *:8883 (LISTEN)
mosquitto 1320 mosquitto 6u IPv6 40356 0t0 TCP *:8883 (LISTEN)

Reply

steve says:

March 16, 2021 at 5:17 pm

Not sure about that.

Bohdan says:

April 16, 2024 at 6:53 am

I have a problem, cannot resolve it for a long time.

I have created client certificates, revoked them. Added to config: crlfile /mypath/to/crlfile.

But mosquitto still allows clients with revoked certificates for some reason.

Also in logs there are no info whatsoever about reading crlfile or something. What do I do wrong?

Reply

steve says:

April 16, 2024 at 8:13 am

I don’t believe it checks for revoked only valid.

rgss

Steve

Reply

Bohdan says:

April 17, 2024 at 7:56 am

From docs https://mosquitto.org/man/mosquitto-conf-5.html:

-crlfile file path

-If you have require_certificate set to true, you can create a certificate revocation list file to revoke access to particular client certificates. If you have done this, use crlfile to point to the PEM encoded revocation file.

Considering this paragraph, i though that mosquitto server would check crlfile by specified path, and block clients with revoked certificates.

So you are saying mosquitto doesn’t work like that?

If it doesn’t work, are there any ways to do that?

Reply

Bohdan says:

April 17, 2024 at 8:09 am

Also found this article: https://primalcortex.wordpress.com/tag/mosquitto/

Look for CRL section

It says it should work

Reply

steve says:

April 17, 2024 at 8:44 am

Haven't used it and didn't realise that the question was aimed at client certificates not sever certificates.
When you think about it the original idea or crl was to block out of date cas which wouldn't be many. In an application using mosquitto and client certificates there could be thousands.
Personally I wouldn't go this route.
On the other side if the serve cert is out of date then it is up to the client to detect it.
As far as I know the Python client doesn't have this logic built it not sure about other clients but doubtful.
Rgds
Steve

Bohdan says:

April 17, 2024 at 12:37 pm

Thanks Steve,

I somehow bumped into a solution. Before i was connecting to local mosquitto via websockets, and when i changed it to mqtt protocol, mosquitto started to block clients with revoked certificates. Couldn't find information yet why it doesn't work for websockets.

Reply

steve says:

April 18, 2024 at 4:16 pm

Good news. How many clients are you trying to block bye the way ?

Reply

keshav says:

February 17, 2021 at 1:52 pm

Hi Steve,

your understanding is impressive Thanks for helping many people's to solve their problems.

I'm facing an issue on TLS. my requirement is to read the key's from the HSM/SoftHSM and pass it to the broker as a key and let broker use that key for TLS.

Currently i'm getting the key's from HSM through java, and not sure how to pass the key to broker, could you please help me out here?

Thanks in advance.

Reply

steve says:

February 17, 2021 at 4:36 pm

Sorry but I'm not familiar with HSM/SoftHSM .

Rgds

Steve

Reply

shri says:

January 26, 2021 at 12:43 pm

Hi Steve,

Looks like a great tutorial with lots of people having it functioning at their ends.

I am working on Windows 10.

I have tried the steps outlined in the tutorial without any success yet. Did anyone got this working on Win 10?

I am using the following:

To run Mqtt broker " mosquitto -v -p 8883 "

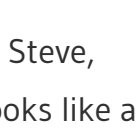
To subscribe "mosquitto_sub -h xxx.xx.xx.x -t test –cafile certs/ca.crt –tls-version tlsv1.2

The moment I give the subscribe command I get the following error on my broker:

1611663604: New connection from xxx.xx.xx.x on port 8883.

1611663604: Client disconnected due to protocol error.

Reply



steve says:

January 26, 2021 at 5:33 pm

Have you tried without using the tls_version switch. Also message seems very short for tls have you checked the config file

rgds

steve

Reply

shri says:

January 27, 2021 at 11:57 am

Hi Steve, Thanks for the prompt reply.

Looks like my config file was somehow not getting picked up. Upon using the -c option with the broker, this started working fine. Below is the command that I used to run my broker.

mosquitto -v -p 8883 -c mosquitto.conf

Reply

Tilen says:

January 25, 2021 at 5:50 pm

Hi!

Great post. I have set up everything “by the book” ☺ but have issues accessing mqtt broker from internet. When trying to access broker from localhost with mosquitto_sub with ssl working fine (with hostname or IP), but when accessing from internet (port forwarding to mqtt server) it keeps getting “Error: A TLS error occurred”. In log I have:

New connection from XXX.XXX.XXX.XXX on port 8883.

OpenSSL Error: error:14094438:SSL routines:ssl3_read_bytes:tlsv1 alert internal error

I am calling with:

mosquitto_sub -h XXX.XXX.XXX.XXX -p 8883 –tls-version tlsv1.2 –cafile ca.crt -t “#”

Any idea? It looks to me like some cert issues.

BR,

Tilen

Reply

steve says:

January 25, 2021 at 6:05 pm

Surprised it works locally using name and ip address.The certificate is ties to the common name. However when that is an issue the error message is usually quite clear. try accessing from another machine locally

Reply

Tilen says:

January 26, 2021 at 8:53 am

Hi,

I tried from another machine in LAN and it works. But as soon I call from imternet (to my public IP and not to LAN IP) gives me error.
I tried with –insecure option and it works, so there must be issue with hostname in certificate.
But I don’t know how to create a certificate that would work also with my public IP (without –insecure option since my client does not have this option).

Thank you in advance,

Tilen

Reply

steve says:

January 26, 2021 at 5:29 pm

Hi

You might be able to use an alternative name but I haven't tried it. Another think to add to my list.

https://serverfault.com/questions/1022573/multiple-ip-adresses-for-a-single-ssl-certificate-no-dns-server

rgds

Steve

Reply

Rida says:

January 11, 2021 at 8:19 am

Hi Steve,

I tried following the tutorial but now my mosquitto broker service won't start up. .

After placing the files in cert folder and changing the .conf file, I start my broker and get the following response:

C:\Program Files\mosquitto>mosquitto -v -p 8883

1610352838: mosquitto version 1.6.12 starting

1610352838: Using default config.

1610352838: Opening ipv6 listen socket on port 8883.

1610352838: Opening ipv4 listen socket on port 8883.

1610352838: mosquitto version 1.6.12 running

After this when start the service it will start up but immediately stop itself.

Reply

steve says:

January 11, 2021 at 1:51 pm

have you tried it using mosquitto -v

Reply

Rida says:

January 12, 2021 at 8:43 am

Yes I have. Same behavior, just the port is changed to 1883.

In my conf file I have:

```
#bind_address

# Port to use for the default listener.
port 8883

and

# certificate files must have ".crt" as the file ending and you must run
# "openssl rehash " each time you add/remove a certificate.
#cafile
#capath
cafile C:\Program Files\mosquitto\certs\ca.crt
keyfile C:\Program Files\mosquitto\certs\server.key
certfile C:\Program Files\mosquitto\certs\server.crt
tls_version tlsv1
```

Reply

steve says:

January 12, 2021 at 2:49 pm

Change the version to 2 and show me the error message

Reply

Rida says:

January 14, 2021 at 1:22 pm

It works.

Thanks. Apparently even after reading the blog over 5 times I somehow missed that line.

I wanted to add one thing, if anyone is trying to run this code on Windows Service. I got the the error message:

'the remote certificate is invalid according to the validation procedure.'

even after adding it to Trusted Root Certification Authorities in User Account. For Self Signed Certificate to work on Windows Service you need to add it as 'Local computer account' for both Trusted Root Certification Authorities and Personal.

Tipsoda says:

January 20, 2021 at 10:49 am

the problem is here:

tls_version tlsv1

it should be

tls_version tlsv1.1 (or 1.2 or 1.3)

Reply

alex says:

January 9, 2021 at 7:28 pm

Hi Steve, I am currently configuring the TLS part mosquitto which I could later use in Paho and I'm having issues, and I am unsure of what my next steps could be.
After generating the certificate, I have placed all the files into one of the desktop folders.
Edited the /mosquitto/conf.d file. The TLS version is edited as 1.2 (in the /mosquitto/conf.d/default.conf) although when I installed openssl it had the (1.1.1d not sure if related but thought it is related). When running netstat -a I was able to see that the port 8883 is in the status LISTEN (although one is in tcp6 and the other in tcp) as well as 9883. However when I follow a mosquitto_sub I get a TLS error occurred.
When generating the certificates, I used the hostname as my IP address and when running mosquitto_Sub I did use an IP for the local host.

My example was like this :
$ mosquitto_sub -h localhost -t "test" –cafile /home/pi/Desktop/ssl/ca.crt
Error: A TLS error occurred.

Perhaps there is an issue of certificates, or firewall? Thank you!
Any tips would be greatly appreciated

Reply

steve says:

January 9, 2021 at 8:50 pm

I try adding the version to the mosquitto_sub command and see if that works
Rgds
Steve

Reply

alex says:

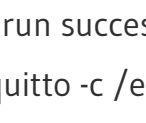
January 11, 2021 at 2:58 pm

Hi Steve, thanks for the reply! Yes I tried it, and added a -d, although I got an error. Steve, if I was to delete the certificates, and create new ones, would the new ones interfere with the old certificates somehow? Because my concern is maybe that the CN on my certificates is not recognised by my raspberry, as my hostname is my raspberrypi and I used for the certificates my IP address. One thing that struck to me was that in one of debug messages, it said – raspberry sending CONNECT. Perhaps that means that the correct CN is the name and not the IP?

$mosquitto_sub -h localhost -t sensor –cafile /home/pi/ssl/ca.crt -p 8883 -d –tls-version tlsv1.2

client mosqsub | 933 – raspberrypi sending connect
OpenSSL error:error:1416F086: SSL routines:tls_process_server_certificate: certificate error failed
Error: A TLS error occurred

Reply

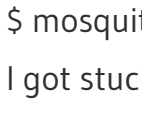


steve says:

January 12, 2021 at 2:46 pm

The certificates wont interfere just put them in a sub folder so you don't mix them up

Reply



Dipankar Maitra says:

January 2, 2021 at 6:48 am

Hi Steve,

Wish you a very happy new year at the beginning. At the same time thanks for your all inclusive page on mosquitto.

I tried all the steps mentioned in http://www.steves-internet-guide.com/mosquitto-tls/ for generating certificates and running mosquitto broker with configuration. and the broker run successfully
$ mosquitto -c /etc/mosquitto/mosquitto.conf -v with the following o/p
1609566743: mosquitto version 1.4.11 (build date 2021-01-01 09:33:00+0000) starting
1609566743: Config loaded from /etc/mosquitto/mosquitto.conf.
1609566743: Opening ipv4 listen socket on port 8883.
1609566743: Opening ipv6 listen socket on port 8883.

But when I tried
$ mosquitto_sub -h localhost -t test -p 8883 –cafile /home/dipadmin/steves/ca.crt,
I got stuck at
Error: A TLS error occurred.
$ mosquitto -c /etc/mosquitto/mosquitto.conf -v throws the following o/p

1609566775: New connection from 127.0.0.1 on port 8883.
1609566775: OpenSSL Error: error:1408F10B:SSL routines:ssl3_get_record:wrong version number
1609566775: Socket error on client , disconnecting.

Also, mosquitto_pub -h localhost -t test -m "Thanks in advance" -p 8883 –cafile /home/dipadmin/steves/ca.crt
Error: A TLS error occurred.

Here are some other input that might help in debugging
OS- Ubuntu 18.04.5 LTS

Openssl version : OpenSSL 1.1.1 11 Sep 2018

```
openssl verify -CAfile ca.crt server.crt
server.crt: OK

**********************************************************************************
*******
$ sudo nano /etc/mosquitto/mosquitto.conf
**********************************************************************************
*******
port 8883

cafile /home/dipadmin/steves/ca.crt

certfile /home/dipadmin/steves/server.crt

keyfile /home/dipadmin/steves/server.key

tls_version tlsv1
**********************************************************************************
********

**********************************************************************************
*********
$ openssl s_client -connect localhost:8883 -CAfile /home/dipadmin/steves/ca.crt
**********************************************************************************
**********
CONNECTED(00000005)
depth=1 C = AU, ST = WBCA, L = KOCA, O = WTCA, OU = IOTCA, CN = diptest01,
emailAddress = dmaitraX@XXXX.com
verify return:1
depth=0 C = IN, ST = WBSR, L = KOSR, O = WTSR, OU = IOTSR, CN = diptest01,
emailAddress = dmaitraX@XXXX.com
verify return:1
—
Certificate chain
0 s:C = IN, ST = WBSR, L = KOSR, O = WTSR, OU = IOTSR, CN = diptest01, emailAddress
= dmaitraX@XXXX.com
i:C = AU, ST = WBCA, L = KOCA, O = WTCA, OU = IOTCA, CN = diptest01, emailAddress =
dmaitraX@XXXX.com
1 s:C = AU, ST = WBCA, L = KOCA, O = WTCA, OU = IOTCA, CN = diptest01, emailAddress
= dmaitraX@XXXX.com
i:C = AU, ST = WBCA, L = KOCA, O = WTCA, OU = IOTCA, CN = diptest01, emailAddress =
dmaitraX@XXXX.com
—
Server certificate
——BEGIN CERTIFICATE——
MIIDizCCAnMCFHOTfXc+rUqNBzVI0DVC0UJoLUxyMA0GCSqGSIb3DQEBCwUAMIGB
MQswCQYDVQQGEwJBVTENMAsGA1UECAwEV0JDQTENMAsGA1UEBwwES09DQTENMAs
G
A1UECgwEV1RDQTEOMAwGA1UECwwFSU9UQ0ExEjAQBgNVBAMMCWRpcHRlc3QwMTE
h
MB8GCSqGSIb3DQEJARYSZG1haXRyYTNAZ21haWwuY29tMB4XDTIxMDEwMjA0Mzgz
```

MFoXDTIxMTIyODA0MzgzMFowgYExCzAJBgNVBAYTAkIOMQ0wCwYDVQQIDARXQINS
MQ0wCwYDVQQHDARLT1NSMQ0wCwYDVQQKDARXVFNSMQ4wDAYDVQQLDAVJT1RTUj
ES
MBAGA1UEAwwJZGIwdGVzdDAxMSEwHwYJKoZIhvcNAQkBFhJkbWFpdHJhM0BnbWFp
bC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDch/rUAQthGsyf
zvQY3IdekfILTzbhC1CGLzTwPGzro85bywDcInsGXJzSij96tBU3ayfbdITUBFtJ
jFHkNRDws2ALPNobDFwNd6dj0T22ohQTMUMrZ3bgEeq26Iyre3I9qX63Ub02wfHM
8UWHYk0QgxvGqNc0b4WfGCjWJpH9plJJZ43LInQ70stdFLKVzKmAP7XgBIpptAbn
oCmWfjw6AbT7VetnIP3JcTphKH82Fv/2NRByYuw0pu0mZ2JntHrI4XbrIU0ZHCVn
IrNyRfdqFsBOSO6gtIeWyBsCZg8GzQ/xvxHofkR8TMEBawQL5+466cID9DfATcRg
AmnytivRAgMBAAEwDQYJKoZIhvcNAQELBQADggEBACY6DmtNOLTJIJ8IBqQGVav2
TJnzTdx0JMyKL6XdQEmdqhMj7ccIeECutJAx/ysYcoQGdHk0S4JGinAYmppNCdcH
M+7gfRVsF01gX8oyEzvYZ7AEiKCc7AR673TcIfcDQEPcOkdkM2B97gbb3Gh2Fz/n
kFMMheh2LHYQg7rIPbA0yTNpKtA67Yt1TviBwd3AmthV1SSV1pi4/yqIzcGFUOAz
L3DnwF9G8sNBpC3ebn29h2c+hVXetiwEJvPVpmwmRzIRzeWTzJw0II850z9JTd8n
vh2fGRcMOF752COUqO/v3Qpmr34mU2S+2hQiPTfM2OWtIizoirg0Oi5If2cekjQ=
——END CERTIFICATE——
subject=C = IN, ST = WBSR, L = KOSR, O = WTSR, OU = IOTSR, CN = diptest01, emailAddress = dmaitraX@XXXX.com

issuer=C = AU, ST = WBCA, L = KOCA, O = WTCA, OU = IOTCA, CN = diptest01, emailAddress = dmaitraX@XXXX.com

—
No client certificate CA names sent
Peer signing digest: MD5-SHA1
Peer signature type: RSA
Server Temp Key: X25519, 253 bits
—
SSL handshake has read 2570 bytes and written 416 bytes
Verification: OK
—
New, TLSv1.0, Cipher is ECDHE-RSA-AES256-SHAServer public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
Protocol : TLSv1
Cipher : ECDHE-RSA-AES256-SHA
Session-ID:
CB1A39D1B43DF7DDC3D0FBBD093584C9BD626AE08A5F4A15EC861104197287BF
Session-ID-ctx:
Master-Key:
7123C09EC3690BA0938A27307A2FBDA9579335D375E3953BDB8890F3014FF7403F8
A3517689498D647547EE5F6F4CF71
PSK identity: None
PSK identity hint: None
SRP username: None
TLS session ticket lifetime hint: 7200 (seconds)
TLS session ticket:

0000 – 1b cc 81 96 a2 1f 2c b9-83 19 41 88 3f a6 0b f9 ……,…A.?…

0010 – 6e 1a f7 42 d3 65 ab 2e-aa 51 f5 3d f7 b2 6a d5 n..B.e…Q.=..j.

0020 – 25 83 ab 18 cd 16 66 02-d6 7f 03 f9 98 84 9d c9 %…..f……..

0030 – 89 57 cd 65 2b e4 c3 94-0e 5e f1 5d f9 86 70 69 .W.e+….^.]..pi

0040 – cb 67 84 24 5f 1e 34 16-80 f1 9f 97 77 80 30 34 .g.$_.4…..w.04

0050 – 44 fe ac 3d 06 27 fd 96-a9 8b 98 ea d6 4e 7b 67 D..=.’…….N{g

0060 – 65 e5 35 88 f3 16 fd b7-d5 8d df 6d d0 27 e9 a9 e.5……..m.’..

0070 – d7 d9 04 ab f5 2e 43 3d-f0 8c e3 0f 2b 3c 9a 40 ……C=….+<.@

0080 – 29 98 4b 79 7d a5 ad 6b-9d 6a 2f 3f 65 ef 45 71).Ky}..k.j/?e.Eq

0090 – 78 e5 a7 f7 63 16 eb b7-34 d2 98 63 c3 c3 c0 9b x…c…4..c….

00a0 – 89 5a 69 c0 af 9a d6 51-ff 7c 2e 99 42 68 53 10 .Zi….Q.|..BhS.

Start Time: 1609565353

Timeout : 7200 (sec)

Verify return code: 0 (ok)

Extended master secret: yes

—

closed

****

****

Please suggest me the next step

Reply

steve says:

January 2, 2021 at 1:55 pm

It looks like it is complaining about the TLS version. Try using the –tls_version option and start at 1.1 using –help wil give you the exact syntax

rgds

steve

Reply

Dipankar Maitra says:

January 3, 2021 at 12:06 pm

Yes Steve .. Thanks you are right

Adding –tls-version tlsv1.2 for mosquitto_sub it started working. Thanks again.

mosquitto_sub -h -t test -p 8883 –cafile /ca.crt –tls-version tlsv1.2

Reply

Paula says:

December 16, 2020 at 12:37 pm

This tutorial is fantastic. However, after following all steps, I was still getting the below error:

SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: IP address mismatch, certificate is not valid for '199.169.9.91'.

The field CN in my server certificate matched the IP of the broker I was connecting to. Still Python complained about the address mismatch. It appears to be that matching the CN to the IP has been deprecated for quite a while and you can have problems depending on your Python version.

I found the solution here:
https://stackoverflow.com/questions/52855924/problems-using-paho-mqtt-client-with-python-3-7
Reply

---

steve says:
December 16, 2020 at 2:57 pm

Hi
The easy way to check for mismatch is to use the insecure option as it doesn't do the check. I wasn't aware of the depreciation and haven't seen anything regarding it until you sent those links.

Reply

Kusvihawan says:
December 14, 2020 at 11:43 am

Hello, this is a wonderful tutorial for a whole section
for this section, I want to ask u something
so, I tried to use this command to pub and i think its work bcs there is no error log:
mosquitto_pub -h KUS -t test -p 8883 –capath /home/user/certs/ca.crt -m "hello"

then i tried this one to subs :
mosquitto_sub -h KUS -t test -p 8883 –capath /home/user/certs/ca.crt
but the message "hello" wont appear
am i did smt wrong? could u help me solve this probs, so curious bout that
btw i run this mqtt broker in the vmware using ubuntu 14.04, the pub and sub in the same machine
and this is the config file:
listener 1883 localhost
listener 8883
certfile /home/user/certs/server.crt
cafile /home/user/certs/ca.crt
keyfile /home/user/certs/server.key

thanks a lot steve ☺
I'm looking forward to ur amazing answer

Reply

---

steve says:

December 14, 2020 at 11:50 am

Are you subscribing in one terminal and publishing in the other?

Rgds

Steve

Reply

Kusvihawan says:

December 14, 2020 at 1:09 pm

yeah, opening 2 terminals 1 for sub and 1 for pubs

is it a mistake?

thanks for the response

Reply

Kusvihawan says:

December 14, 2020 at 2:37 pm

yep, am i not supposed to do that?

pls i need ur help to solve this

thanks ☺

Reply

steve says:

December 14, 2020 at 5:15 pm

NO you need two terminals and you need 1 to subscribe first before you publish. Can you confirm it works without ssl.

rgds

steve

Reply

Kusvihawan says:

December 15, 2020 at 3:00 am

i think it works well without SSL because i can see the message on the sub's side

thanks for ur response ☺

steve says:

December 15, 2020 at 1:24 pm

Do you have access to the broker console. If so you should be able to see the messages being published to the broker and from the broker.

rgds

steve

Kusvihawan says:

December 15, 2020 at 3:34 am

is it possible to use port 8883 or other than 1883 without SSL?

steve says:

December 15, 2020 at 1:23 pm

Yes you can use any port you want as long as no other process is using them

---

Kusvihawan says:

December 16, 2020 at 4:36 am

actually, i can get the process log without ssl using mosquitto -v this includes the message and any package like connack and suback
but idk with SSL bcs the output is just like :
1608092046: mosquitto version 1.6.3 starting
1608092046: Config loaded from /etc/mosquitto/conf.d/kon.conf.
1608092046: Opening ipv4 listen socket on port 1883.
1608092046: Opening ipv4 listen socket on port 8883.

and no other output after that
Sorry for asking too much i hope u r ok with this ☺

steve says:

December 16, 2020 at 9:02 am

you need to start mosquitto using the -v option to see all messages

---

ram says:

November 7, 2020 at 1:07 pm

mosquitto terminal:
1604753903: New connection from 192.168.0.102 on port 8883.
1604753903: OpenSSL Error[0]: error:14094412:SSL routines:ssl3_read_bytes:sslv3 alert bad certificate
1604753903: Socket error on client , disconnecting.
python server:
File “/build/iotmaster/iotmaster/wsgi.py”, line 32, in
from iotdasbrd import cloudmqtt
File “/build/iotmaster/iotdasbrd/cloudmqtt.py”, line 298, in
mqttc.connect(“192.168.0.102”, 8883, 60)
File “/home/mgk/.local/lib/python3.8/site-packages/paho/mqtt/client.py”, line 941, in connect
return self.reconnect()

File “/home/mgk/.local/lib/python3.8/site-packages/paho/mqtt/client.py”, line 1104, in reconnect
sock.do_handshake()
File “/home/mgk/.local/lib/python3.8/site-packages/eventlet/green/ssl.py”, line 311, in do_handshake
return self._call_trampolining(
File “/home/mgk/.local/lib/python3.8/site-packages/eventlet/green/ssl.py”, line 161, in _call_trampolining
return func(*a, **kw)
File “/usr/lib/python3.8/ssl.py”, line 1309, in do_handshake
self._sslobj.do_handshake()
ssl.SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: IP address mismatch, certificate is not valid for ‘192.168.0.102’. (_ssl.c:1123)
config file:
Place your local configuration in /etc/mosquitto/conf.d/
#
A full description of the configuration file is at
/usr/share/doc/mosquitto/examples/mosquitto.conf.example
pid_file /var/run/mosquitto.pid
persistence true
persistence_location /var/lib/mosquitto/
#log_dest file /var/log/mosquitto/mosquitto.log
include_dir /etc/mosquitto/conf.d
port 8883
cafile /etc/mosquitto/ca_certificates/ca.crt
keyfile /etc/mosquitto/certs/server.key
certfile /etc/mosquitto/certs/server.crt
tls_version tlsv1.2
i am using paho client, what was the problem?

Reply

steve says:

November 7, 2020 at 5:43 pm

It looks like the certificate name you are using is incorrect. The common name that you set on the certificate must match the name used to access the mqtt broker. In your case it should be 192.168.0.102.
Use the insecure option on setup and it doesn’t perform this check and should work provided there are no more errors.

Reply

Mario M. Brenes says:

September 30, 2020 at 12:19 am

It is possible to skip the self-signed certificate as shown in this thread.
https://pi3g.com/2019/05/19/python-paho-mqtt-client-self-signed-certificates-websockets-howto/

Reply

steve says:

September 30, 2020 at 9:23 am

Looks like it but I haven't tried it.

Rgds

Steve

Reply

Truc says:

September 10, 2020 at 1:52 pm

Above you said that "shouldn't use encryption (-ds3)"

But in detail you said "Note: it is OK to create a password protected key for the CA.".

So what I have to do ?

Reply

steve says:

September 11, 2020 at 2:51 pm

Don't use password protected if you aren't sure.

Reply

Dave says:

September 10, 2020 at 3:05 am

Hi Steve,

I have a ev ssl certificate signed by entrust and the .csr was generated from IIS, windows. I retrieved the private key from the certificate manager and used Root.crt as cafile and the signed certificate.crt as certfile. However, I am getting this error on the broker -> OpenSSL Error[0]: error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown this error when I try to connect my client (with .pfx) to my broker.

Reply

steve says:

September 11, 2020 at 3:05 pm

Hi

Try and create your own cert and keys and get it working then move back to the entrust ones once you are happy with the procedure.

rgds

steve

Reply

Bach says:

May 23, 2021 at 3:25 am

Hi,
I don't know specific what you use the broker for, but i have the same error and in my case, i have fixed by this:

listener 8883
protocol mqtt

cafile C:\Program Files\mosquitto\certs\ca.crt
certfile C:\Program Files\mosquitto\certs\server.crt
keyfile C:\Program Files\mosquitto\certs\server.key

listener 9883
protocol websockets

cafile C:\Program Files\mosquitto\certs\ca.crt
certfile C:\Program Files\mosquitto\certs\server.crt
keyfile C:\Program Files\mosquitto\certs\server.key

Reply

Syed Ali Hassan says:

October 27, 2022 at 6:39 am

Hi,
I'm facing the same problem. I have the same configuration. On the other hands my open port and websockets are working fine. I just receiving an error on MQTT secure port 8883. Error detail are written below

OpenSSL Error: error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown
Socket error on client , disconnecting.
New connection from on port 8883.
OpenSSL Error: error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown
Socket error on client , disconnecting.
|– mosquitto_auth_acl_check(..., client id not available, <jwt_token)

Reply

steve says:

October 27, 2022 at 12:17 pm

There seems to be something wrong with the certificate you are using. What client do you use to test.
rgds
steve

Reply

Supriya says:

September 9, 2020 at 10:51 am

Hi Steve.
Your articles are amazing and have helped me many times!!

I have a ubuntu server configured with ip x.x.x.x and i have installed mosquitto broker here. Also I followed the steps to configure TLS from this article.
Now my client is an ubuntu desktop with ip y.y.y.y and I have copied the ca.crt file from my broker to this machine.
When I run the python script i get "Unable to connect : TLS error occured " Also the script gives me "Socket error"
What am I doing wrong?
The mosquitto config file is same as your's . Please help!

Reply

steve says:

September 9, 2020 at 3:30 pm

have you tried using the mosquitto_pub tool.
Rgds
Steve

Reply

Supriya says:

September 10, 2020 at 5:44 am

Yes. I tried to publish with certificate using mosquitto_pub. I get the TLS error.

Reply

steve says:

September 11, 2020 at 3:06 pm

use the ask steve page to contact me and then you can send me the cert and keys via email
rgds
steve

Reply

Marwan says:

August 13, 2020 at 5:33 am

This is a very helpful tutorial, Steve, Thanks so much.
I followed the steps you explained and I was successful running the broker with the TLS options. However, I had a problem connecting clients to the broker using mosquitto_sub/mosquitto_pub commands. when I run:
mosquitto_pub -t “test” –cafile mqtt-ca.crt -m “HELLO THERE ON THE OTHER SIDE” -h mqtt-broker
I get: Unable to connect (Lookup error.). on the client side and:
1597295923: New connection from 127.0.0.1 on port 8883.
1597295923: OpenSSL Error[0]: error:14094438:SSL routines:ssl3_read_bytes:tlsv1 alert internal error
1597295923: Socket error on client , disconnecting.
On the server side (They are actually the same PC).

However, using the –insecure option works fine. Tha same for mosquitto_sub. I believe I have a problem in the host name but I don’t know how to fix it. In the certificate signature requests (for both CA and server) I used the same common name “mqtt-broker”. I also tried two different CNs for CA and server certs but I got the same output error.

I also tried connecting from another PC in the network and the same scenario happened.

Thanks,
Marwan.

Reply

steve says:
August 13, 2020 at 3:18 pm

The name you need to use is the name you use to connect to the broker. So on a local network it may be mqtt-broker.local.
if you can ping the broker using
ping mqtt-broker
then it should work but you are correct that the -insecure means a naming issue.
rgds
steve

Reply

Marwan says:
August 13, 2020 at 6:35 pm

Thanks so much, I found out that the CN should be the same as the PC name itself (it was a stupid of me). So, when I renamed my PC to mqtt-broker, the connection was successful without the –enable option but in my case it was mqtt-broker.fios-router.home as the hostname. I think I have to do some work on my router.

Many thanks!

Reply

steve says:

August 14, 2020 at 5:43 pm

Well done

Reply

Brian says:

August 3, 2020 at 8:51 pm

Steve, props to the wonderful tutorials you provide for MQTT functionality. These helped me more than everything else on the web.

This system was working perfectly fine when I was using 9001 port with ws then ……. SSL had to come into play (policies).

BUT….I am running into a problem with the SSL setup and connecting to the broker via WS for my webapp.

1. Mosquitto.config
#start (default) listener on port 1883
port 1883

#start listener on port 8883 with SSL
listener 8883
certfile /etc/mosquitto/certs/……pem
cafile /etc/mosquitto/ca_certificates/……pem
keyfile /etc/mosquitto/certs/…….key

listener 8083
protocol websockets
certfile /etc/mosquitto/certs/……pem
cafile /etc/mosquitto/ca_certificates/……pem
keyfile /etc/mosquitto/certs/…….key

2. The following command works and sends
mosquitto_pub -h test.site -t smth/smth –cafile /etc/mosquitto/ca_certificates/….pem -m “test” -p 8883

3. The following command does not work
mosquitto_pub -h test.site -t smth/smth –cafile /etc/mosquitto/ca_certificates/….pem -m “test” -p 8083
ERROR- A network protocol error occurred when communicating with the broker.

4. For my webapp I am using MQTT package with React

This will not connect
import React from ‘react’;

```
import ‘./index.css’;
const mqtt = require(‘mqtt’)

const websocketUrl = “wss://test.broker:8083”

var options={
clientId:”random”,
rejectUnauthorized : false,
ca: ‘./…….crt’ (in client same as used above)
}

const client = mqtt.connect(websocketUrl, options)
```

As I said above the 9001 worked with ws none SSL site and now this change is not working.
Might you have any ideas what I am doing wrong?
Reply

steve says:
August 4, 2020 at 2:09 pm

I don’t think the mosquitto_pub tool supports websockets. Ytu using mqttbox which is a chrome extension as it support websockets with ssl
Rgds
Steve

Reply

Brian says:
August 7, 2020 at 3:11 pm

Thanks for your input Steve! I tried MQTTBox and it is for sure an interesting tool. I will get more in depth with it later.

To test the Mosquitto side of things I use MQTT-Explorer and the server allows connection on all ports I configured in the Mosquitto.conf file including the SSL secured ports. For the SSL secured ports you simply add the CA cert in MQTT-Explorer within the advanced settings area and it connects with no problems.

My problem is that the MQTTjs library for some reason will not connect to the SSL port client side to my MQTT broker. I think it has to do with the formatting of the CA cert I am giving the library to work with, but I am not for certain since the same format was used in MQTT-Explorer. I need to figure out what format the library is requiring.

Reply

steve says:
August 7, 2020 at 3:26 pm

Hi
what error messages do you get? I assume you are using the nodejs client is that correct?
rgds
Steve

Reply

kiran says:
September 23, 2020 at 4:36 am

I am getting this below error at client side:
Client mosq-8EelCay0nUa53G4DIA sending CONNECT
OpenSSL Error[0]: error:14094418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca
Error: A TLS error occurred.
i just copied server CA certificate to client but not signed the client certificate with it.
my client certificate is signed with its own ca certificate.
i am using below command.
mosquitto_pub –cafile –cert –key –insecure …
do i need to sign client certificate with copiedCA certificate ? if yes is there any way to avoid this(i used –insecure option still same problem)
i also tried multiple combination for this command but i think probem is with ca certifciate only at client side.

Reply

steve says:
September 23, 2020 at 1:38 pm

Hi
Have you seen this tutorial
http://www.steves-internet-guide.com/creating-and-using-client-certificates-with-mqtt-and-mosquitto/

Does plain ssl work that is no client certificates
rgds
steve

Reply

Kiran says:
September 25, 2020 at 10:27 am

Hi Steve,
Thanks for the response.
Yes i saw that tutorial and yes plain ssl is working.

l also tried by setting require_certificate flag to false in mosquitto.conf at broker and in this case client is validation server correctly.
so this scenario is working fine.
but i want client validation at server/broker side for which i need to set require_certificate flag to true (as per mosquitto.conf man page).
But when i set require_certificate to true at broker side, i am getting error: “tlsv1 alert unknown ca”
i have just copied CA certficate of broker to client and passing it to command mosquitto_pub –cafile
The thing is we dont want to copy server/broker CA key at client, we can just copy server/broker CA.crt to client.

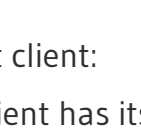
FYI:
At server/broker:
1. broker has its own ca so server_ca.crt, server_ca.key and from this CA cert signed server.crt, server.key
mosquitto.conf at broker
cafile server_ca.crt
certfile server.crt
keyfile server.key
require_certifcate true
(no other flags are set here; i tried setting use_subject_as_username/use_identity_as_username but still same problem)

At client:
client has its own ca so client_ca.crt, client_ca.key and from this CA cert signed client.crt, client.key
in addition to that CA certificate(server_ca.crt) copied server/broker

and from client hitting below command:
mosquitto_pub -d -p 8883 -h -m “Hello” -t test –repeat 10 –cafile –cert client.crt –key client.key

getting error:
sending CONNECT
OpenSSL Error[0]: ……. :tlsv1 alert unknown ca

Regards
Kiran
Reply



steve says:
September 25, 2020 at 6:20 pm

The client ca should be the same as the server ca.Try using my scripts and create some new keys and see if that works any better
rgds
steve

Shiva says:

July 25, 2020 at 3:13 am

Hi Steve,
i m using mqtt node js client to connect with same configuration as you mentioned here, but what i have observed is i'm able to connect to broker with any client certificate. And when i change the configuration to required_certificate : true. , i m getting this error : error:1417C0C7:SSL routines:tls_process_client_certificate:peer did not return a certificate.

Reply

steve says:

July 25, 2020 at 2:13 pm

That is probably correct as until you set the require certificate the broker doesn't check them. If you enable require certificate then you need a valid one.
Rgds
Steve

Reply

Aravinda Kumar says:

July 12, 2020 at 10:33 pm

Hi steve,
I am getting this error "OpenSSL Error: error:14094418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca".

pub command executed:
mosquitto_pub -h 127.0.0.1 -t "test_subscribe" -p 8883 -m "hi" –cafile "/etc/mosquitto/certs/m2mqtt_ca.crt"

My .conf file:
listener 1883

listener 8883
cafile /etc/mosquitto/certs/m2mqtt_ca.crt
keyfile /etc/mosquitto/certs/m2mqtt_srv.key
certfile /etc/mosquitto/certs/m2mqtt_srv.crt

listener 8083
protocol websockets
cafile /etc/mosquitto/certs/m2mqtt_ca.crt
certfile /etc/mosquitto/certs/m2mqtt_srv.crt
keyfile /etc/mosquitto/certs/m2mqtt_srv.key

the common names point to 127.0.0.1 and i am using linux
Can you help me out please?

Reply

steve says:

July 13, 2020 at 7:23 pm

Try using the –insecure option and if it works then it is a problem with the ca name. If not then copy the ca.crt file into your local folder and try again as it maybe a permissions problem.

Reply

torntrousers says:

July 11, 2020 at 8:52 am

Thanks for the great tutorial.
I'm trying to use an intermediate certificate to sign client certificates but can't get it to work, do you know if thats possible?
So ca.crt signs the mqtt server.crt and ca.crt signs intermediate.crt which signs client.crt and then conactenate the intermediate.crt and client.crt into a clientbundle.crt

Reply

steve says:

July 12, 2020 at 8:11 am

Hi
Should be possible but I've never done it.

Reply

Stefano says:

July 2, 2020 at 10:11 am

Hi Steve, thanks for this brilliant tutorial!
Any clue why the certificates generated for CN=127.0.0.1 would give rise to:

ssl.SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: IP address mismatch, certificate is not valid for '127.0.0.1'. (_ssl.c:1108)

on the client side?

Thanks for your time!

Reply

steve says:

July 2, 2020 at 1:17 pm

Hi
You need to use either the IP address of the broker or the domain name as the common name on the certificate and the client has to use this when it connects to

the broker.

So if you use the ip addess then the client has to connect with the iP address.

Rgds

Steve

Reply

Tim T says:

July 8, 2020 at 3:05 pm

I created a ca.crt and a server.crt with both CN: 127.0.0.1 . Then I started a Broker on my osx. But always when I try to connect with `mosquitto_pub -t test/ -m “hi” –cafile ./ca.crt -h 127.0.0.1 -p 8883` I get the error:

OpenSSL Error[0]: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed

Error: A TLS error occurred.

The broker says:

1594220727: OpenSSL Error[0]: error:14094418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca

1594220727: Socket error on client , disconnecting.

Reply

Tim T says:

July 8, 2020 at 4:20 pm

Okay it’s caused by using the same CN for ca.crt and server.crt

Reply

Go says:

July 2, 2020 at 7:50 am

Hi Steve,

thanks a lot for you tutorials,

do you think it’s possible to communicate between mqtt and react-native with SSL?

I follow you tutorial about SSL and I success to establish a communication between my python client and mqtt but not with react-native in android device.

thanks in advance if you can help me with SSL between broker mqtt and react native.

My configuration:

// web sockets configuration

listener 9001

websockets protocol

cafile /usr/local/etc/mosquitto/certs_ws/ca.crt

keyfile /usr/local/etc/mosquitto/certs_ws/server.key

certfile /usr/local/etc/mosquitto/certs_ws/server.crt

require_certificate true // doesn’t work with true or false in android

listener 8883

protocol mqtt

cafile /usr/local/etc/mosquitto/certs_mqtt/ca.crt

keyfile /usr/local/etc/mosquitto/certs_mqtt/server.key

certfile /usr/local/etc/mosquitto/certs_mqtt/server.crt

my broker is installed in raspberry pi 4

Reply

steve says:

July 2, 2020 at 8:38 am

Hi

Sorry but I've never worked with react native. But I would suspect that it is an SSL issue and you need to add the ca to a certificate store or try without SSL.

Reply

Go says:

July 2, 2020 at 11:57 am

without SSL it's working.

I don't success to add ca on certificate store

Reply

steve says:

July 2, 2020 at 1:18 pm

Take a look here it may help

https://www.lastbreach.com/blog/importing-private-ca-certificates-in-android

Reply

Go says:

July 2, 2020 at 3:25 pm

Thanks Steve,

I will try this solution.

if I can't do it and you know someone who can make my request, I'm ready to pay it to make me an industrial solution that allows SSL to work with reac_native_mqtt lib.

I can create a upwork project or in a other website working development .

Thank you

HSN says:

June 23, 2020 at 8:50 pm

One of the best article about creating certificates!

Reply

Mohamed Ilies Boudouma says:

June 16, 2020 at 12:06 pm

hello Steve,

What is the differnce between CA cert & self signed cert ?

some client tools I use like MQTTBox uses self signed and it worked

Reply

steve says:

June 16, 2020 at 12:22 pm

Hi

Take a look at this

https://cheapsslsecurity.com/blog/self-signed-ssl-versus-trusted-ca-signed-ssl-certificate/

Reply

ASIF KHAN PATHAN says:

June 11, 2020 at 9:55 am

You are just amazing.

Reply

Sherry Wang says:

April 29, 2020 at 3:54 pm

Hi Steve,

Thank you for all these helpful information about this subject. I am trying to run the mosquitto broker and client on the local machine with SSL. I have followed your instructions to create the CA certificate, server certificate and the server key. I placed these files in the folder and changed the configuration file accordingly as below:

cafile C:\mosquitto\certs\ca.crt

certfile C:\mosquitto\certs\server.crt

keyfile C:\mosquitto\certs\server.key

port 8883

tls_version tlsv1

Then I restart the mosquitto broker. However, following test failed:

mosquitto_pub -h 9XLMZY2 -t test/topic –cafile C:\mosquitto\certs\ca.crt -m “Hello” -p 8883

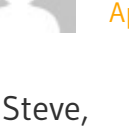
The error message is “Error: No connection could be made because the target machine actively refused it.”.

But, when I try following test, it success.

mosquitto_pub -t test/topic -m “Hello”

Seems the configuration is not taking effect. The broker is still working at non-SSL mode. What I have done wrong?

Thanks a lot.
Sherry
Reply



steve says:

April 29, 2020 at 4:32 pm

Hi
That error message is common when the port is blocked by a firewall or not open on the target machine.
Are you running mosquitto from the command line? When testing I always run mosquitto from my home folder and use the -c switch to load the configuration file e.g
mosquitto -c ssl.conf
that way you can see the console and know straight way if the ports are open
rgds
steve

Reply

Auggie Li says:

April 19, 2020 at 11:36 pm

Hi Steve,

I followed your page to create the keys for connections between Flutter and Ejabberd, and copied ca.crt to client side. But I am getting the following errors for iOS, but it is good on Android.

flutter: Socket Connection failed: HandshakeException: Handshake error in client (OS Error: CERTIFICATE_VERIFY_FAILED: ok(handshake.cc:354))

It seems the verify is ok, but it got some errors. Is it because it is self signed? For more details of my questions, please visit https://stackoverflow.com/questions/61220693/mqtt-between-flutterejabberd-tls-handshake-exception-on-ios-not-android-while.

Thanks,
Auggie

Reply

steve says:

April 20, 2020 at 4:34 pm

If it works ok on ANdroid then it is unlikely to be a problem with self signed. It could be an SSL version problem on IOS But I don't use Apple and so can't check it.
Rgds
Steve

Reply

Auggie Li says:

April 25, 2020 at 11:39 pm

Thank you Steve. You are right, it is very likely a SSL version problem. Even though I still cannot figure out how it works on IPhone X simulator, it can work on my physical iPhone 6S, which is good enough for me. Thank you for your answer, otherwise I would have wasted much time on looking at self-signed. ☺

Reply

Fiqih Prawida says:

March 19, 2020 at 2:49 pm

Hello Steve,
Thanks for this tutorial, I have tried this step and successfully.
I Have some questions:
1. Does each client need to be made a certificate?
2. How can I create a certificate for each client?

Thanks

Reply

steve says:

March 19, 2020 at 2:54 pm

The clients all use the the same CA certificate
rgds
steve

Reply

Ram says:

February 25, 2020 at 11:41 am

Hi,
I created the tls certificate as per your tutorial. while trying run A TLS error occurred.
mosquitto_pub -h localhost -t 'test/topic' –cafile /home/pi/Documents/iotmaster/ca.crt -m 'helloWorld' -p 1883

ERROR:Unable to connect (A TLS error occurred.).

this is my config file

```
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /var/run/mosquitto.pid

#persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

#include_dir /etc/mosquitto/conf.d
port 1883
#listener 1883
cafile /etc/mosquitto/certs/ca.crt
keyfile /etc/mosquitto/certs/server.key
certfile /etc/mosquitto/certs/server.crt
#tls_version tlsv1
```

Reply

steve says:

February 25, 2020 at 2:58 pm

Hi

Do you see anything on the mosquitto console?

Rgds

Steve

Reply

Ram says:

February 26, 2020 at 6:44 am

thanks for your reply

In console i get this error

Unable to connect (A TLS error occurred.).

Reply

steve says:

February 26, 2020 at 5:08 pm

Sorry

I mean on the mosquitto console.

Rgds

Steve

Reply

ping says:

July 7, 2020 at 5:17 pm

Hi,

I have the exact problem as you, did you solve it now ?

Reply

ashsampree says:

February 18, 2020 at 8:50 am

Hello Steve,

I actually configured Mosquitto to work over TLS but PUB/SUB is only working for localhost only.Can you please help me out in PUB/SUB using another IP address.I work on Ubuntu virtual machine please help me out?

Reply

steve says:

February 18, 2020 at 10:32 am

Hi

Have you copied the ca.crt file to the remote machine?

rgds

steve

Reply

Noah says:

February 14, 2020 at 6:37 am

Thanks for the tutorial, you have made things much clearer!

I was following your explaination and i think it should work fine, but somehow mosquitto does not recognize the ca.crt file.

1581661924: mosquitto version 1.6.8 starting

1581661924: Config loaded from /mosquitto/config/mosquitto.conf.

1581661924: Opening ipv4 listen socket on port 8883.

1581661924: Opening ipv6 listen socket on port 8883.

1581661924: Error: Unable to load CA certificates. Check cafile "/home/pi/docker/mosquitto/config/ca.crt".

1581661924: Error: No such file or directory

On my raspi I tried to "sudo nano /home/pi/docker/mosquitto/config/ca.crt" and of course i could open it.

Any ideas, why mosquitto has these problems?

Additionally: The ca.crt ca I use for all my clients, correct? So if mosquitto runs on the Raspi, I use the ca.crt to access with MQTTfx and also copy the certificate into my esp8266 code?

Thanks!!

Reply

steve says:

February 14, 2020 at 9:17 am

It looks like a permission problem on the file or folder.

rgds

steve

Reply

Girish Kumar says:

January 2, 2020 at 5:48 pm

Hi

I am using paho client on Raspberry PI to connect to a mosquito broker.

My code to connect is as follows:

```
def mySens(sensorid,subscriberID):
clientID = sensorid
client = mqtt.Client(client_id=clientID)
client.on_connect = when_connect
client.on_message = on_message
client.username_pw_set("xxxxxxx","yyyyyyyyy");
client.tls_set('ca.crt')
x = client.connect(host, port)
print(x, host,port)
flag = True
while(flag == True):
x = client.publish(topic="MASTER/HELLO", payload="hello")
x = client.publish(topic="DEVICE/WELCOME", payload=json_string)
client.loop_forever()
```

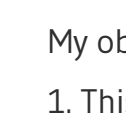
When x = client.connect(host, port) executes I get the following error

```
Exception in thread figure01
Traceback (most recent call last):
File "/usr/lib/python3.5/threading.py", line 914, in _bootstrap_inner
self.run()
File "/usr/lib/python3.5/threading.py", line 862, in run
self._target(*self._args, **self._kwargs)
File "sim.py", line 35, in sensorsimulator
x = client.connect(host, port)
File "/home/pi/.local/lib/python3.5/site-packages/paho/mqtt/client.py", line 937, in
connect
```

return self.reconnect()
File "/home/pi/.local/lib/python3.5/site-packages/paho/mqtt/client.py", line 1100, in reconnect
sock.do_handshake()
File "/usr/lib/python3.5/ssl.py", line 996, in do_handshake
self._sslobj.do_handshake()
File "/usr/lib/python3.5/ssl.py", line 641, in do_handshake
self._sslobj.do_handshake()
ssl.SSLError: [SSL: SSLV3_ALERT_HANDSHAKE_FAILURE] sslv3 alert handshake failure (_ssl.c:720)

Tried googling not able to find out the root cause or a solution. _ Can you help me ?

Reply



steve says:

January 2, 2020 at 6:34 pm

I noticed you used
client.username_pw_set("xxxxxxx","yyyyyyyyy");
client.tls_set('ca.crt')
are you using authentication and certificates? If so have you tried without them
Rgds
Steve

Reply

Girish Kumar says:

January 6, 2020 at 5:08 am

Hi Steve,
Thanks for your response. I had disabled the password based authentication and tested only with certificate and the problem is still there.
My observations.
1. This problem is seen only with Raspberry Pi, on windows the same python program which uses paho library is working fine with Certifcate and Password based authentication
2. Same is working on ESP8266 with password and certificate.
3. I When I disable the certificate and use only password based authentication the it works on Raspberry Pi. But I cannot use as the user name and password are transmistted as clear text in MQTT.
For the deployment I Am working, I need to User name and password plus, TLS .

My guess is the TLS library with RPi is having a bug

Regards
Girish

Reply

steve says:

January 6, 2020 at 9:27 am

Hi

I would agree. Try upgrading the library.

Rgds

Steve

Reply

Selin Demir says:

December 16, 2019 at 5:49 pm

Hi Steve,

Thanks for all these helpful informations about this subject. I use a broker and a publisher on same machine, Raspberry Pi and have a subscriber on Windows machine. I followed your descriptions and it worked fine in command prompt.

I can also publish with python script on Raspberry and get the message on Windows command prompt. (C:\Program Files\mosquitto>mosquitto_sub -h 192.168.1.104 -t konu –cafile certs/ca.crt -p 8883)

But my subsriber.py can not see the message despite using tls.set() method. I see an error like this:

self._sslobj.do_handshake()

ssl.SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: IP address mismatch, certificate is not valid for '192.168.1.104'. (_ssl.c:1076)

Here is my subscriber.py:

```
#-*-coding:utf-8-*-
import paho.mqtt.client as mqtt

def on_message(client, userdata, msg):
print(msg.topic + " " + str(msg.payload))

def on_disconnect(client, userdata, rc):
mqtt.connect("192.168.1.104", 8883, 60)
#mqtt.connect("mqtt.eclipse.org", 1883, 60)

def on_connect(client, userdata, flags, rc):
mqtt.subscribe("konu")

mqtt = mqtt.Client()
#mqtt.tls_insecure_set(True)
mqtt.tls_set("c:/Program Files/mosquitto/certs/ca.crt",tls_version=2)
mqtt.on_disconnect = on_disconnect
mqtt.on_connect = on_connect
mqtt.on_message = on_message
mqtt.connect("192.168.1.104", 8883, 60)

mqtt.loop_forever()
```

Note: I had a copy of ca.crt which I created on Raspberry C:\Program Files\mosquitto\certs\ca.crt on Windows

Thank you so much again i'll be waiting for your return.

Reply

steve says:

December 16, 2019 at 6:34 pm

Hi
It is because you are using the ip address and not the name that is on the certificate
uncomment this line.
#mqtt.tls_insecure_set(True)
If it works then that is the reason
rgds
steve

Reply

Selin Demir says:

December 16, 2019 at 7:56 pm

Step 4 :You could use the IP address or Full domain name. You must use the same name when configuring the client connection.

As you mentioned above i used my broker's ip adress as common name on step 2 and step 4. I also uncomment the line you said but it didnt work. What should I do now? I appreciate your help..

Note: I want you to remind that it worked fine for command prompt but it doesn't work with subscriber python script.

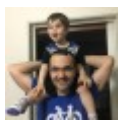
Reply

steve says:

December 16, 2019 at 8:48 pm

Hi
remove the tls version here
mqtt.tls_set("c:/Program Files/mosquitto/certs/ca.crt",tls_version=2)
Can you use the ask steve page if you still have errors and we can deal with it with email as it s easier
rgds
steve

Reply



Gazi says:

December 22, 2019 at 8:40 pm

This works:

https://stackoverflow.com/a/52856943/679553

Reply

G Muthu says:

November 18, 2019 at 2:35 pm

Hi,

I created the tls certificate as per your tutorial. while trying mosquitto_pub –cafile /etc/mosquitto/certs/ca.crt -p 8883 -h 192.168.237.201 -t ‘test’ -m “tstmsg” –insecure I am getting A TLS error occurred.

Could you help me to resolve this problem.

This is my configuration file.

persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

#port 1883
#listener 1884

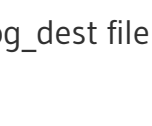
port 1883
listener 8883

require_certificate true

#tls_version tlsv1.1

cafile /etc/mosquitto/certs/ca.crt
keyfile /etc/mosquitto/certs/server.key
certfile /etc/mosquitto/certs/server.crt

Reply



steve says:

November 18, 2019 at 6:33 pm

Hi

Remove the line

require certificates true

Rgds

Steve

Reply

G Muthu says:

November 20, 2019 at 2:05 pm

Hi steve,

Thanks for the help.

I am able to connect using python with ssl. In case of java able to connect normally but not with tls., can you refer some sample application for java.

Reply

steve says:

November 20, 2019 at 3:51 pm

Hi
Sorry but I've never used Java.
Rgds
Steve

Reply

Pradeep says:

November 2, 2019 at 2:17 pm

Have setup mosquitto on AWS ES2 and on the same machine mosquitto_sub without -cafile connection is fine, with inclusion New connection from 11.22.33.44 on port 8883. 1572703961: Client disconnected due to protocol error Any help would be appreciated. Followed all above said steps, and in this case ca.crt would be the same fine..

Reply

steve says:

November 3, 2019 at 5:55 pm

Usually with SSL it is a wrong ca file or a common name mismatch. Check that you are using the correct ca file
rgds
steve

Reply

pradeep says:

November 7, 2019 at 5:42 am

Thanks for the response, tested with broker and client on the same machine, this means same ca.crt file, still the same error. Without SSL the setup is working just fine,

Regards
Pradeep

Reply

steve says:

November 7, 2019 at 6:14 pm

Use the ask steve page and send me your conf file Also what client are you using?

Reply

Silvanu says:

May 7, 2020 at 2:41 pm

I had same issue and this was caused by “listener” in mosquitto.conf file. I have replaced “listener 8883” with “port 8883” and it worked.

Reply

Dian says:

September 21, 2019 at 9:07 am

hey steve, thank you for this tutorial
i want to ask how to subscribe the topic from other device using mosquitto mqtt tls? i try to add command “–cafile certs\ca.crt” to subscribe, but it doesn’t work
cause i want to subscribe from other device
I’ve install and configuring tls too from other device, but it doesn’t work to subscribe

Reply

steve says:

September 21, 2019 at 9:26 am

Hi
What client are you using?Does it work without SSL?
Rgds
Steve

Reply

Dian says:

September 22, 2019 at 4:49 am

If subscribe without tls its can work normally, but I want to subscribe using mqtt tls (secure mqtt),
sorry my english so bad

Reply

steve says:

September 22, 2019 at 10:47 am

Can you use the ask steve page and send me the commands you are using to publish and subscribe and a screen shot of the mosquitto console error.
http://www.steves-internet-guide.com/ask-steve/
Rgds
Steve

Reply

Dian says:

September 23, 2019 at 4:19 am

I use rasp pi for publisher and PC for subscriber,
If sending message without tls it work normally , but in this problem I want to sending message with TLS,
Can you tell me, how to subscribe for other device?
Sorry my english so bad

Reply

---

steve says:

September 23, 2019 at 8:25 am

Hi
What client are you using and is the broker setup for SSL?
Rgds
Steve

Reply

Wayne says:

September 18, 2019 at 11:40 am

Hi Steve,
I really enjoy your tutorials and insight to the MQTT topic. How would this certificate process be different if you want to use an F5 load balancer to offload the TLS workload?
Thanks,
-Wayne

Reply

---

steve says:

September 18, 2019 at 3:33 pm

Glad You find the tutorials useful but sorry I can't offer any insights into the question as I'm not really involved with load balancing.
Rgds
Steve

Reply

Minh says:

September 10, 2019 at 7:35 am

Hi Steve.
I use vitural server AWS EC2 and mosquitto , when i create CA key , i put the common name (random , ex mytest) , in Server.crt I put common name is public DNS of server , but when i test with MQTTFx it not working with error:
1568099939: New connection from 113.161.92.36 on port 8883.
1568099939: Socket error on client , disconnecting.
And MqttFx show MQTTException.
How can i fix it?
Regards!

Reply

steve says:

September 10, 2019 at 10:01 am

Hi
Have you tried using mosquitto_pub tool?
You would need to send me your files and access details for me to take a look. You can use the ask steve page.
Rgds
Steve

Reply

kaouther says:

August 16, 2019 at 4:27 pm

Thank you so much for the great article. Beside SSL or username/Password authentification can I use other authentification factors? if it's possible how can I modify the mosquitto.conf file ? thank you in advance.

Reply

steve says:

August 16, 2019 at 5:51 pm

Hi
Plugins are available but I haven't used them yet. Take a look here
https://github.com/iegomez/mosquitto-go-auth
rgds
steve

Reply

Hung says:

July 30, 2019 at 4:33 am

Hi Steve,

Thanks a lot for all your great articles on MQTT.

I followed your instructions, except the commen name in step 2 and step 4 is I use the ip address. Everything is fine, I check on MQTT.fx quite perfectly, but when I check on MQTT.box it is not very good.

On MQTT.box, I only set up the "HOST" as the server's ip address and "SSL/TLS Certificate Type" the type is: CA signed server certificate. I haven't been able to issue CA.crt yet, it can connect. Can you explain help me? Thank you very much.

Reply

steve says:

July 30, 2019 at 2:23 pm

Not sure as I don't use either of those tools. It may be using port 1883. Take a look to see if it is enabled on the broker.

Rgds

Steve

Reply

Thangz says:

July 23, 2019 at 6:59 am

If I am running mosquitto on 'localhost', can I use the same (localhost) for my server certificate common name?

Each help would helpful.

Best Regards,

Thangz

Reply

steve says:

July 23, 2019 at 8:14 am

Hi

The name you use is the name you would use when you ping the machine from another machine on the network.

Because many home/test networks don't use dns then you could use the ip address or if it is a windows network the computer name.

For a test network you can also tell the client to ignore the common name which isn't secure but it isn't a problem on a test network

If you use the name localhost it will not work correctly from another machine.

Rgds

Steve

Reply

Leon H says:

July 5, 2019 at 8:15 am

I am trying to connect with TLS 1.2 to CloudMqtt Broker which I can do w/o a problem when no security protocol involved…(using M2Mqtt library)

You said that:

——-

In this case we only need a trusted server certificate on the Client.

We do not need to create client certificates and keys but this is covered in Creating and Using Client Certificates with MQTT and Mosquitto

——-

So how the MqttClient constructor should look like?

I tried this and it goes through… but later the Connect call throws communication exception:

X509Certificate caCert = X509Certificate.CreateFromCertFile(mCaServerCertFIle);

//X509Certificate clientCert = X509Certificate.CreateFromCertFile(clientCertFile);

mqttClient = new MqttClient(serverProfile.ServerAddress,

port,

true,

caCert,

null,//clientCert,

MqttSslProtocols.TLSv1_2,

RemoteCertValidationCallback

//LocalCertSelectionCallback

);

}

Reply

steve says:

July 5, 2019 at 8:28 am

Leon

You only need to use the ca from cloud mqtt which is on your machine as it is a public ca.

However you can also download it as I needed to do with Python. here is what they say:

https://www.cloudmqtt.com/docs/faq.html#TLS_SSL

———

How do I connect using TLS (SSL)? Where do I find cert and key files?

If you connect by TLS/SSL, add –capath or –cafile and point it to a cert store. Our server cert is signed by Comodo, which has the AddTrust CA as root. Most OSs comes with it by default, so can you point to your default trust/CA store. (example: –cafile=/etc/ssl/certs/ca-certificates.crt) If you don’t have a trust store you can

download the AddTrust/Comodo root cert from
https://support.comodo.com/index.php?/Default/Knowledgebase/Article/View/979/108/domain-validation-sha-2

More information can be found here, under Certificate based SSL/TLS Support. You also need to use the port for MQTT over TLS (see above).
=====
I would download it and then get it to work that way.
Let me know how you get on
Rgds
Steve
Reply

Dioris Moreno says:
June 26, 2019 at 4:22 pm

Great article Steve, thanks for sharing!

Reply

RK says:
May 28, 2019 at 10:57 am

After the fifth step this is the error I am getting:
unable to load CA Private Key
1995601392:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt:../crypto/evp/evp_enc.c:536:
1995601392:error:0906A065:PEM routines:PEM_do_header:bad decrypt:../crypto/pem/pem_lib.c:439:
Why does this error occur?

Reply

steve says:
May 28, 2019 at 4:30 pm

Is this error occurring when you execute the command in step 5?
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 360

rgds
Steve

Reply

RK says:
May 29, 2019 at 7:00 am

Yes the error occurs after the execution of the command.

Reply

steve says:

May 29, 2019 at 8:15 am

Hi
You probably made an error in an earlier step. The easiest thing is to start again and see if it works.
Rgds
Steve

Reply

Tam Nguyen says:

April 9, 2019 at 11:35 am

Hi Steve, thank you so much about SSL posts. I have a question about hostname, I use IP address for CA, and I know the hostname CA need to match. But have a problem, my address is dynamic and it can't match anytime. You said we could use –insecure to ignore checking it with python, but I use Node-red on each server. How can I pass it without checking the hostname, thank you so much!

Reply

steve says:

April 9, 2019 at 12:36 pm

Hi
There is an option on the ssl settings of the mqtt node called verify server certificate. This is the same as the insecure option.

Reply

Henrik says:

April 2, 2019 at 3:14 pm

Great tutorial. Could the client certificate be created from the server certificate instead of from the CA certificate (if we want to be able to generate certificates dynamically on the server but not store the CA key there)?

Reply

steve says:

April 3, 2019 at 8:03 am

Don't know but I wlll check.
Rgds
Steve

Reply

fabio says:

February 13, 2019 at 10:22 pm

Hello Steve,
first of all thank you for your work. I followed every step and installed mosquitto on a raspberry pi (jessie). At the moment I can't get it to work, so ask for your help.
Some information:
– my broker certificate common name is fsMQTTbroker and my CA certificate common name is fsCA
– my raspi has manually assigned IP (192.168.1.9)
– the internet connection works
– my mosquitto.conf is the following:
d_file /var/run/mosquitto.pid
persistence true
persistence_location /var/lib/mosquitto/
log_dest file /var/log/mosquitto/mosquitto.log
port 8883
cafile /etc/mosquitto/ca_certificates/ca.crt
keyfile /etc/mosquitto/certs/server.key
certfile /etc/mosquitto/certs/server.crt
tls_version tlsv1

The problem is:
– when testing with
mosquitto_pub -h fsMQTTbroker -t topic/example –cafile /etc/mosquitto/ca_certificates/ca-crt -m "test" -d
i get the error "Unable to connect (Lookup error.)."
– when I try
mosquitto_pub -h fsMQTTbroker -t topic/example –cafile /etc/mosquitto/ca_certificates/ca-crt -m "test" -d
i get the error
Client mosqpub|2261-raspberryp sending CONNECT
Error: host name verification failed.
OpenSSL Error: error:14090086:SSL routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
Error: A TLS error occurred.

every suggestion will be useful.

thanks,
fabio

Reply

steve says:

February 14, 2019 at 9:46 am

Didn't spot a difference in the commands used.
The name or ip used to connect to the broker must match the common name on the certificate otherwise you get that error.
There is a –insecure option that tells the client to ignore that check. Use this to see if it works

Reply

MQTT Novice says:
January 31, 2019 at 5:59 pm

hey! great tutorial but i didn't get it to work. Not sure what I am doing wrong…
My set up is using MQTT.Fx to access the broker @ test.mosquitto.org.
Questions:

1- Lets say the name of my pc is "My-PC" and the server I want to connect to is "test.mosquito.org". Would I enter "My-PC" in the common name for CA.crt and enter "test.mosquitto.org" in the common name for server.crt??

2- when asked to add extra fields (optional password etc.) I just press enter and move along. I don't bother putting a period there…is that ok?

3- Then when I have completed generating CA and signing the server certificate, I would open MQTT.Fx and use the signed ca.crt with the broker address "test.mosquitto.org" and use "My-PC" as the client name. Is that correct?

Thanks for looking!

Reply

steve says:
January 31, 2019 at 6:49 pm

1.The common name of the server.crt should match the domain name of the server that it is installed on.
The name of your PC isn't important and not used on the certificates. The common name for the Ca would be usually be a company name.
2. Hitting enter should be ok
3.yes but the client name could be anything.

Reply

VagiaM says:
December 17, 2018 at 12:05 pm

Hi Steve,
Your tutorial is excellent!
I have followed all the steps and it feels that everything is going well. However, I have a problem that I am trying to resolve.

When I run your script to check the paho client I get the following error:
"Traceback (most recent call last):
File "Desktop/PahoClient1.py", line 21, in
client1.tls_set('/Home/Downloads/Python-3.6.1/mqtt-demos/ca.crt', tls_version=2)
File "/usr/local/lib/python2.7/dist-packages/paho_mqtt-1.4.0.dev0-py2.7.egg/paho/mqtt/client.py", line 772, in tls_set
context.load_verify_locations(ca_certs)
IOError: [Errno 2] No such file or directory"

I have installed Python-3.6.1, while python 2.7 was already installed on Ubuntu. Does the cause of the error is that I installed paho client on Python-3.6.1 and not on python 2.7?

Thank you in advance!

Vagia
Reply

---

steve says:
December 17, 2018 at 1:31 pm

Tks for the nice comment
It could be. When you have multiple versions of Python when you do a PIP install it might get installed for the new version.
You need to check if the mqtt client is installed for 3.6 which you can do by using
pip show paho-mqtt
to see where pip will install use
pip –version.

Can you confirm that you can pub/sub without ssl?
You also need to check the location of you ca file when using ssl
You might find this useful
http://www.steves-internet-guide.com/python-notes/
I created it because I had the same problems when I started.
Let me know how you get on
rgds
steve

Reply

Alexander says:
November 23, 2018 at 8:36 pm

Hi Steve!
THANK YOU for this really cool howto! It works great!
I have a question:
When i verify and sign the server certificate with "-days 360", does this mean, that i have to update the files on the clients physically every 360 days?
Please don't lought, my client are 100 km away. I don't want to go there by car.

I'm the absolute dud on such server things, so ...

Greetings from austria!

Reply

steve says:

November 24, 2018 at 10:46 am

Yes you are correct and tks for pointing it out I will add some notes to the tutorial.

You need to create the certificates with a long expiry time.

The article below uses -days 365000

https://securityboulevard.com/2018/03/creating-long-term-ssl-certificates/

rgds

steve

Reply

Mo says:

November 14, 2018 at 7:33 am

Steve,

When apply this: openssl req -new -x509 -days 1826 -key ca.key -out ca.crt

I get some error like this:

——————————————

Enter pass phrase for ca.key:

Can't load ./.rnd into RNG

7240:error:2406F079:random number generator:RAND_load_file:Cannot open file:cryp

to\rand\randfile.c:88:Filename=./.rnd

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]:CN

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

Locality Name (eg, city) []:ShenZhen

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

Organizational Unit Name (eg, section) []:Pro T&V

Common Name (e.g. server FQDN or YOUR name) []:Yongxiang

Email Address []:Yongxiang@mo.com

——————————————————————————-

Looks like I can get the ca.crt, but How can I resolve the error?

I use the laptop do this, OS is win7 64BIT

Reply

steve says:

November 14, 2018 at 9:19 am

Hi
I haven't seen the error. Does it happen every time? Does the certificate still work?

Reply

David Peters says:
October 29, 2018 at 7:59 pm

Has anyone been able to solve the "tlsv1 alert unknown ca" message? I've been through the tutorial several times and can not find out what is causing the problem.
Client:
Error: A TLS error occurred.

Server:
1540843163: New connection from xxx.xxx.xxx.xxx on port 8883.
1540843163: OpenSSL Error: error:14094418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca
1540843163: OpenSSL Error: error:140940E5:SSL routines:ssl3_read_bytes:ssl handshake failure
1540843163: Socket error on client , disconnecting.

Reply

noyreddine says:
November 8, 2018 at 11:11 am

have you fixed it ?

Reply

jan says:
October 29, 2018 at 2:27 pm

Hey,
I justed wanted to say thanks for the tutorial. It was the only one the net which let me enable secure communication successfully ☺ I think the other instructions fail to mention how important the domain is. I added the local ip address and it worked just fine. I also tested it with home assitant and it worked ☺

So, a big thank you for this!

best regards

Reply

wayne says:
October 23, 2018 at 1:40 am

Hi Steve , just to clarify ,from a comment made by vicky on 11 April 2018 , it was mentioned that if the input field of step 2 and step 4 were identical , the ca.crt would not work ,hence only the server.crt is applicable. If the input fields of step 2 and 4 were of different values , it means that the ca.crt would work fine without error?Further more if i am using mosquitto library , can the ca.crt and server.crt certificates be used? Or is there a need to create a pem version?

Regards
Wayne

Reply

steve says:

October 23, 2018 at 1:30 pm

Yes if the forms ca.crt and server.crt are identical then it can cause problems.
The server.crt is not applicable as it need a ca.crt on the client so you would be best to create the ca.cr and server.crt again.
You don't need a pem version as you aready have one as pem is encoding and I 'm pretty sure the tutorial create a pem version. See this guide
http://info.ssl.com/article.aspx?id=12149%20
Rgds
Steve

Reply

wayne says:

October 25, 2018 at 6:45 am

Hi, steve
thanks for explaining , so i can assume that ca.crt = ca.pem? i googled and some people are using the .pem and i am getting a little confused
Rgds,
Wayne

Reply

steve says:

October 25, 2018 at 9:17 am

Yes and no. The easiest way is to open the certificate in a text editor and if you can read it and it starts with BEGIN CERTIFICATE the it is in perm format.
I was as well confused by all of the different extensions as there isn't really a fixed relationship.
On your own setup I would stick to the .crt and .key extensions which seem more common.
If you have a ca.pem certificate then you can rename it to ca.crt and it will work as normal.

rgds

steve

Reply

wayne says:

October 27, 2018 at 3:28 am

Hi, thanks for explaining once again.Followed your guide and it work . However ,when i tried using openssl (s_client -connect domainname:8883 -showcerts) to test the connectivity, i was return with an error, “Verification error: self signed certificate in certificate chain” . Can i seek your opinion and guidance on this?

Regards,

Wayne

steve says:

October 27, 2018 at 7:02 pm

Hi

It’s normal and ok see here

https://stackoverflow.com/questions/12180552/openssl-error-self-signed-certificate-in-certificate-chain

It should still work ok

Rgds

steve

umit says:

October 3, 2018 at 7:12 am

Hi steve

I have a question about mqtt security at all. As we know MQTT designed the way that any client who subscribe to a topic can receive the messages that publish on that topic. So a question is that in a big network with lots of users and devices connected to that how should we prevent a user to sniff or publish messages to other users devices? for example we have user1 with device1 belonging to that and user2 with device2. for example we have a topic for device1 that lets user1 control it, how should we prevent user2 which is connected to that broker to publish or subscribe to device1 topics? considering there’s lots of devices and users.

Thanks in advance

Reply

steve says:

October 3, 2018 at 8:21 am

Any security you will need to build into the clients like using access tokens etc. However the brokers also provide various degrees of security like ACLs and

username and passwords.
http://www.steves-internet-guide.com/topic-restriction-mosquitto-configuration/
http://www.steves-internet-guide.com/mqtt-username-password-example/
Personally I like message encryption as it is end to end see here
http://www.steves-internet-guide.com/encrypting-the-mqtt-payload-python-example/
Reply

umit says:
October 4, 2018 at 2:17 pm

Thanks for your articles and answering questions. Those were great. I agree with you, message encryption is a better way to go but we should also use TLS if we want to secure the topics as well. Also considering that ACLs and username and passwords need another service to control the brokers resources
Reply

steve says:
October 4, 2018 at 4:45 pm

You could always use encrypted topic names as well to hide the actual names.
rgds
steve

Reply

umit says:
October 9, 2018 at 10:40 am

Yes that's a good way. Just a problem is that if some one sniffs the packets, can find the topic and start to send huge data to that topic and cause the devices not to work properly.

steve says:
October 9, 2018 at 4:05 pm

To prevent that you can use Access control lists and username/password authentication
see
http://www.steves-internet-guide.com/topic-restriction-mosquitto-configuration/
http://www.steves-internet-guide.com/mqtt-username-password-example/

Richard says:
September 25, 2018 at 8:45 pm

If you're getting "Socket error on client , disconnecting." you should look in your config if allow_anonymous is set to False. In this case, using certificate, set it to True or

provide username during logging.

Reply

Peri says:

August 31, 2018 at 8:57 am

I was using certificate generation process mentioned on CentOS 7. Configured mosquitto for websockets, when starting mosquitto broker, I am getting 'OpenSSL doesn't support ECDH' error.

Reply

steve says:

August 31, 2018 at 2:13 pm

Sorry I'm not familiar with CentOS 7

Reply

Peri says:

September 1, 2018 at 4:03 pm

Oh... Okay.

Reply

sachin says:

August 30, 2018 at 6:34 pm

Please explain the significance of ca.crt and server.crt....I am not able to distinguish. Justify more please.

Reply

steve says:

August 31, 2018 at 2:13 pm

Both are certificates. You can consider them as the same as passports.
The server certificate contains the public keys for that server.
The CA certificate contains the public keys of the certificate authority which can be self signed or signed by an higher certificate authority.
The ca private signature key is used to sign the server certificate. It is the trusted authority.
When a client connects to a server to use SSL the server sends the client its certificate which contains its public key (which has been signed by a CA (trusted authority) and the client uses the public signature key in the CA certificate to verify that the server public key is valid.
For this to work the client must have a copy of the CA certificate.
CA certificates for public certificate authorities like verisign are included with your

browser.

Does this make sense?

Reply

sachin says:

September 3, 2018 at 4:19 am

OK got it ..Thank you. Is it possible to provide client side authentication using MQTT ? I read to your reply ,it says – YES but complexity is more. I want to know whats the complexity inovolved?

Reply

steve says:

September 4, 2018 at 4:56 pm

The complexity is generating and distributing the keys for each client.

Reply

Christoph says:

August 14, 2018 at 1:59 pm

Hey Steve, thanks for this great tutorial, it was my starting point in securing my mosquitto broker communication. Just a question, do you think its a good idea to buld and use for each client a different private key (ca.crt)? Additionally I want to use user-id and password autentification.

Kind regards,
Christoph

Reply

steve says:

August 14, 2018 at 4:53 pm

The Ca.crt is the certificate authority certificate and Usually you only use 1 for all your clients.
You can use certificate authentication which means giving each client it's own key but It would probably be too difficult to manage and I haven't tried it.
The tutorial for username password is here
http://www.steves-internet-guide.com/mqtt-username-password-example/
I would recommend getting ssl to work then getting username./password to work and then combine them at the end
rgds
steve

Reply

Cristofer says:

August 8, 2018 at 3:19 pm

hi steve.

mosquitto can use .jks files for ssl security?

in this tutorial you only use .crt files

Reply

Sahithya Kodam says:

July 20, 2018 at 7:30 am

Hi Steve,

Really helpful article. I followed all the steps listed but I am receiving an error that says "Error: Problem setting TLS options". The command I am running is mosquitto_sub -t home/livingroom -v -d –cafile ca_certificates/ca.crt -h 192.168.0.32 -p 8883. The CN on both CA and server certificate is 192.168.0.32. I also tried using the option –tls-version tlsv1.

My mosquitto.conf file has the following contents

port 8883

cafile /etc/mosquitto/ca_certificates/ca.crt

keyfile /etc/mosquitto/certs/server.key

certfile /etc/mosquitto/certs/server.crt

tls_version tlsv1

Can you point out where I could be going wrong?

Thank you.

Reply

---

steve says:

July 20, 2018 at 8:25 am

Try the full path for the certificate file and also try the –insecure option and comment out the tls_version in the conf file otherwise looks ok

rgds

steve

Reply

Stefan says:

May 14, 2019 at 12:54 pm

Im my case I had to put the mqtt version as well:

mosquitto_sub -V mqttv311 -h 192.168.178.31 -p 8883 -t “test” –cafile /etc/mosquitto/ca_certificates/ca.crt

and

mosquitto_pub -t “test” -m “hi2” –cafile /etc/mosquitto/ca_certificates/ca.crt -p 8883 -h “raspberrypi” –insecure -V mqttv311

(@Steve: Great tutorial!)

Reply

Jan says:

June 28, 2018 at 8:02 am

In log i get error: Error: Unable to load CA certificates. Check cafile “/root/jbre/SSL/ca.crt”

I i comment out server.key it loads mosquitto or if i comment out ca.crt, mosquitto works, so i guess those two files are not compatible….hm… i did generate keys with step 2 and 4 with slightly different value… but i also leave some fields empty, like mail, maybe thats problem?

Reply

---

steve says:

June 28, 2018 at 8:57 am

Problems with the certificates will show up when you try to connect. If it doesn’t load then Check the paths.
Send me your 3 files steve@steves-internet-guide.com and I’ll test them
rgds
steve

Reply

Jan says:

June 27, 2018 at 1:37 pm

Hey.
First, really nice and useful blog. So i did everything exactly like you wrote in this configuration but when i try to connect with client to broker i get “Error: Connection refused”. I am trying simple with “mosquitto_sub -d -v -h 20.0.0.211 –insecure –cafile /home/ubuntu/jbre/SSL/ca.crt -t test -p 8883”.

Maybe any idea?
Thank you in advance.

Reply

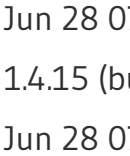
---

steve says:

June 27, 2018 at 3:31 pm

Connection refused is often when you use the wrong port or IP address the command you are using looks OK I would check the broker.

Reply



Jan says:

June 28, 2018 at 7:12 am

Thank you for fast answer. I see now that when i add in mosquitto.conf cafile, certfile, keyfile mosquitto broker can't start or is in failed status:
Jun 28 07:07:19 kibernetmq mosquitto[1776]: 1530169639: mosquitto version 1.4.15 (build date 2018-05-05 12:54:33+0000) starting
Jun 28 07:07:19 kibernetmq mosquitto[1776]: 1530169639: Config loaded from /etc/mosquitto/mosquitto.conf.
Jun 28 07:07:19 kibernetmq mosquitto[1776]: 1530169639: Opening ipv4 listen socket on port 8883.
Jun 28 07:07:19 kibernetmq systemd[1]: mosquitto.service: main process exited, code=exited, status=1/FAILURE
Jun 28 07:07:19 kibernetmq systemd[1]: Unit mosquitto.service entered failed state.
Jun 28 07:07:19 kibernetmq systemd[1]: mosquitto.service failed.

When i comment this lines out mosquitto starts normally and is active:
Jun 28 07:06:46 kibernetmq mosquitto[1766]: 1530169606: mosquitto version 1.4.15 (build date 2018-05-05 12:54:33+0000) starting
Jun 28 07:06:46 kibernetmq mosquitto[1766]: 1530169606: Config loaded from /etc/mosquitto/mosquitto.conf.
Jun 28 07:06:46 kibernetmq mosquitto[1766]: 1530169606: Opening ipv4 listen socket on port 8883.
Jun 28 07:06:46 kibernetmq mosquitto[1766]: 1530169606: Opening ipv6 listen socket on port 8883.

Any idea?

Thanks again,

Reply

steve says:

June 28, 2018 at 7:18 am

The most likely cause is that it can't find one of the files or there is a syntax error in the conf file.
When testing I would start the broker manually from the command line using mosquitto -c myconfile.conf.
Place the myconfile in the home directory as it is easier than having to edit the conf file in the etc folder.
You can move it there when done.
Rgds
Steve

Reply

Jan says:

June 28, 2018 at 7:19 am

Btw in mosquitto.conf i only have:
port 8883
cafile /root/jbre/SSL/ca.crt
certfile /root/jbre/SSL/server.crt
keyfile /root/jbre/SSL/server.key
require_certificate true

Reply

steve says:

June 28, 2018 at 7:29 am

comment out the require_certificate line it is for client certificates

Victor Praxedes says:

June 15, 2018 at 5:29 pm

Some stuff i found out following your tutorial:
– Common name MUST be you computer name. I couldn't find out how to use wildcards to make it work on a PC in a domain (PC at work), but on a non network managed computer (my home computer) it finally worked.

– If you set 'tls_version tlsv1' in the mosquitto.conf file, you MUST use '–tls-version tlsv1' on the pub/sub command line or it will default to TLS v1.2
> mosquitto_sub -h DESKTOP-09SCS82 -p 8883 –cafile ca.crt -t hello/world –tls-version tlsv1

Reply

Victor Praxedes says:

June 19, 2018 at 2:19 pm

Quick update, got it working on a managed network. Here's the deal, your System window (windows key + pause break) has three informations:
– Computer Name
– Full Computer Name
– Domain

You should use "*.your.domain.name.com" as your CN so you can estabilish a connection from every computer in the network that has the certifcate. When stabilishing a connection, your host must be the "Full Computer Name" information like mosquitto_sub -h PC023.your.domain.name.com -p 8883 –cafile ca.crt -t hello/world –tls-version tlsv1

Reply

Sam says:

October 5, 2018 at 9:02 pm

Thank you so much! This solved my issue.
Common name is so important. When generating Certificate Sign Request, we have to use "*.myDomain.com" or "myComputerName.myDomain.com" as the common name. Then when doing the client connection, host has to be "myComputerName.myDomain.com".
That is how the TLS certificate works!

Reply

Leave a Reply

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

Comment *

Name *

Email *

Website

Post Comment