Updated:January 2, 2021 / By steve

Configure a Mosquitto Bridge With SSL Encryption

It is very likely that a bridged connection between two brokers will be encrypted.

The Mosquitto broker (server) provides two methods of using **SSL encryption** on a bridged connection

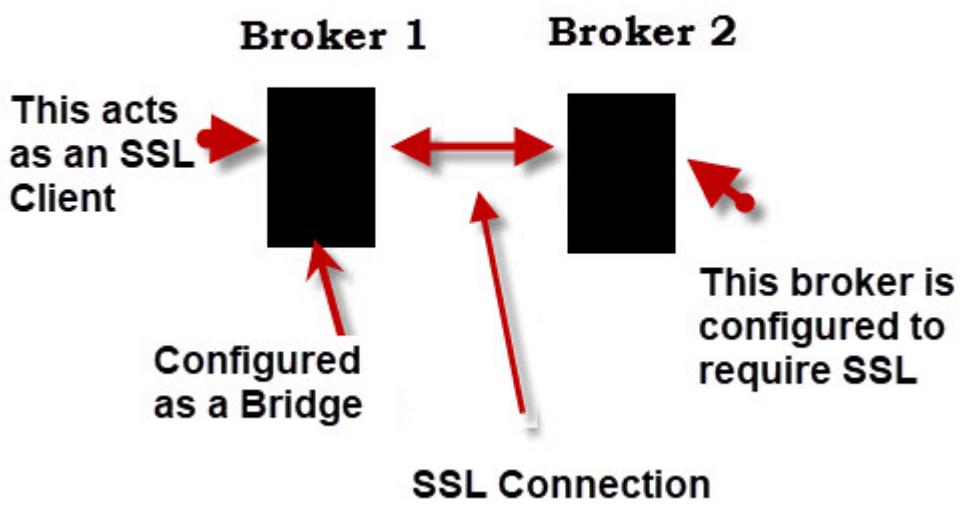
- Certificate encryption
- PSK encryption

In this tutorial we will be configuring a secure bridged connection using both methods.

If you are new to certificates then you should read this tutorial on SSL encryption and certificates before continuing.

Broker Setup Overview

in this tutorial we will bridge topics on broker 1 to broker 2.



Polls

How Do You monitor your brokers (up/down)

- Don't monitor
- manually
- Own Scripts or node-red flows
- Python scripts from this site
- Node-red flow and dashboard
- Other

Vote

View Results

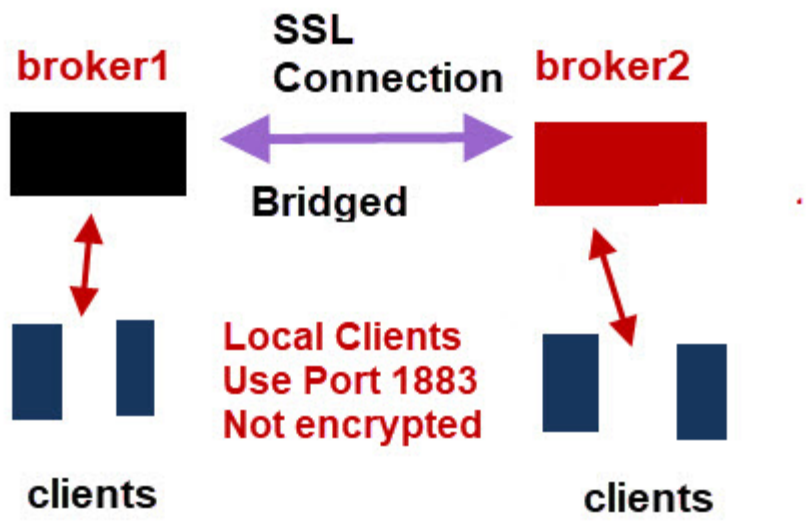
Email Newsletter

Join me Up

Broker 1 will be configured as bridge and is effectively an SSL client.

broker 2 will operate as a normal broker, and will not require any configuration for bridging. It will act as an SSL server.

Generally locally connected clients will use the standard port 1883 and not use encryption as shown in the diagram below:



SSL Encryption Using Certificates

Broker 2 needs to be configured as an SSL server and require encryption. I've chosen to use **port 8883.**

Notice the configuration is for an **extra listener**, and not for a bridge.

Here is the relevant part of the config file on broker 2 showing the SSL settings.

```
# listener port-number [ip address/host name]
# listener 9001
# protocol websockets
listener 8883
protocol mqtt
cafile c:\mosquitto\certs\ca.crt
keyfile c:\mosquitto\certs\server.key
certfile c:\mosquitto\certs\server.crt
```

Under extra Listeners section

Now **broker1** needs to be configured as a bridge.

The setup is almost identical to a normal bridge connection except we need to add a line for the **CA file** and also change from using an IP address (192.168.1.184) to a name (**ws4**).

This is because my **server key** on broker 2 was generated with the name **ws4**. See the Mosquitto ssl tutorial for details.

Here is the bridging part of the config file:

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

Search

Search ...

- Buy Me A Coffee
- About Me
- MQTT Tools
- Networking

My Youtube Channel

- node-red
- MQTT Brokers
- mqtt and python
- Internet

```
#connection <name>

connection bridge-01
#address 192.168.1.184:1883
address ws4:8883
bridge_cafile /etc/mosquitto/certs/ca.crt
topic # out 0 "" b1/
topic # in 0 "" b1/
```

uses name and ssl port

certificate authority file

Note: No **server key** is needed on broker 1 as it is **functioning as an SSL client.**

Testing

The easiest way of testing is to create an error which you can easily do by commenting out the encryption setting on broker 1

You should get an SSL error on broker 1

PSK Encryption Overview

The mosquitto broker supports PSK encryption which can be used instead of certificate based encryption.

> *In cryptography, a pre-shared key (PSK) is a shared secret which was previously shared between the two parties using some secure channel before it needs to be used. –Wiki*

This is the same type of encryption used on Wi-fi Networks.

The key used in Mosquitto is restricted to hexi decimal numbers i.e 0-9,A,B,C,D,E,F

You can generate the key using online **key generators**, random number generators or just make one up.

For testing purposes it is easier to **make one up**.

For real world deployments a security policy would need to be created and used.

Note: PSK encryption uses **SSL** just like certificate based encryption.

P**SK encryption** isn't supported on the Paho Python client, and so cannot be used to encrypt a client broker connection.

Configuring PSK on a Mosquitto Bridge Connection

Using the same setup as before. **Broker1** is configured as a bridge and **broker2** is a normal broker.

There are two settings that you need to add to **broker2**

- **psk_hint**
- **psk_file**

The **psk_hint** option is very important as this is what tells the broker to use PSK.

The **actual value** that you enter doesn't appear important for mosquitto but may be in other PSK implementations.

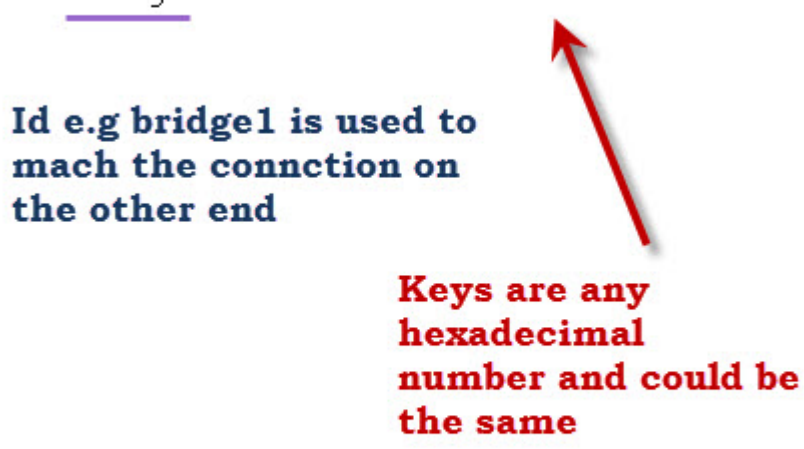
There can only be one **psk_file** entry.

Below is sample configuration file:

```
port 1883
log_type all
listener 8883
psk_hint my test bridge
psk_file c:\mos\certs\psk_file.txt
#cafile c:\mos\casas\ca-sas.crt
#keyfile c:\mos\casas\sas-server.key
#certfile c:\mos\casas\sas-server.crt
```

The contents of the **PSK file** are shown below:

```
bridge1:1234567899876543 21
bridge2:1234567899876543 22
```



Broker 2 PSK Config File

Note the above file is for two PSK connections our current connection will use **bridge1**.

Broker1 is the bridge and here is the configuration:

Bridge Configuration for PSK and SSL

```
listener 1883
listener 8883
log_type all
connection bridge-01
address sas:8883
bridge_identity bridge1
bridge_psk 12345678998765432l
#bridge_cafile /home/pi/mos/certs/ca-sas.crt
topic # both 0
```

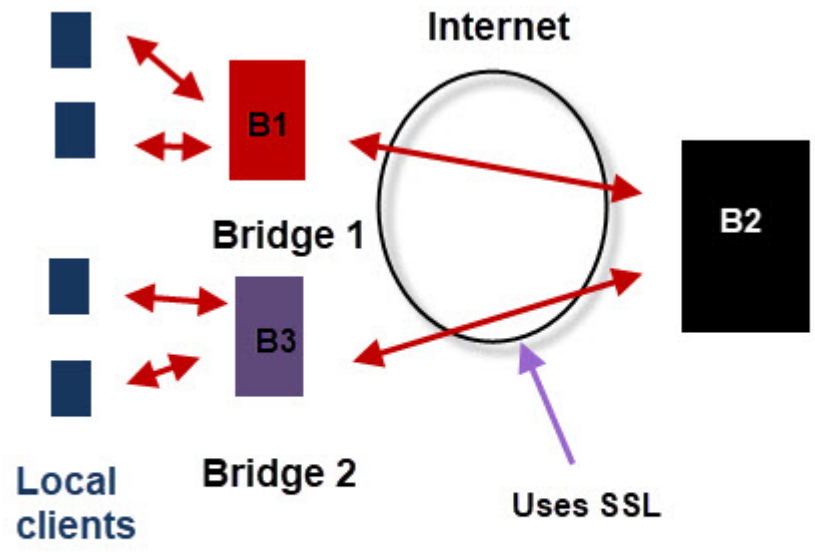
The important entries are the bridge identity **bridge1** which matches the bridge identity in the PSK file on broker2.

The **bridge_psk** value matches the one in the PSK file on broker2.

Multiple Bridge Connection Examples

We will now examine two configuration scenarios. We will use PSK for SSL but the same applies if using certificates.

The diagram below depicts two bridge connections. This would be typical central broker with branch offices configuration



Configuring Multiple Bridged SSL Connections

Broker2 needs no configuration changes to support multiple bridged connections for both certificate based and PSK.

However it may need additional entries in the PSK file. The **psk file** shown previously is already configured for two connections.

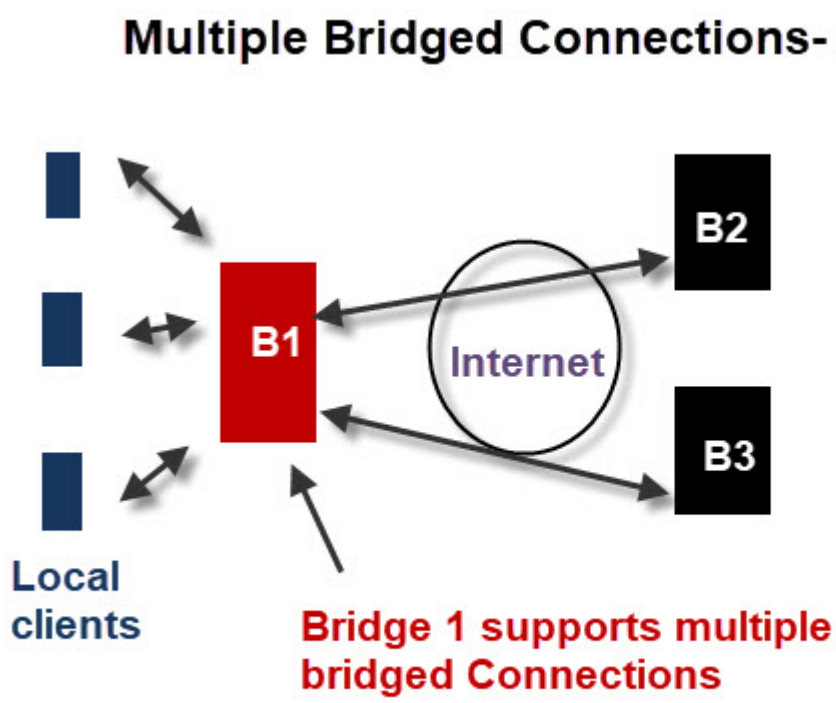
The configuration on broker 1 (bridge 1) is that shown previously and needs no changes

The configuration file for B3 (bridge2) is shown below:

```
listener 1883
listener 8883
log_type all
connection bridge-02
address sas:8883
bridge_identity bridge2
bridge_psk 12345678998765432 2
#bridge_cafile /home/pi/mos/certs/ca-sas.crt
topic # both 0
```

Multiple Bridged Connection -Example2

This time we will configure the bridge to have multiple bridged connections. This would be a branch office to central broker with redundancy and is depicted in the diagram below:



Broker 1 Configuration

We would usually use the same port for each bridge so we only need a single listener.

Each bridge connection starts with the connection name.

Below we see two connections called **bridge-01** and **bridge-02.**

Here is the configuration file

```
listener 1883
listener 8883
log_type all
connection bridge-01
address sas:8883
bridge_identity bridge1
bridge_psk 12345678998765432l
#bridge_cafile /home/pi/mos/certs/ca-sas.crt
topic # both 0

connection bridge-02
address sas:8883
bridge_identity bridge2
bridge_psk 12345678998765432 2
#bridge_cafile /home/pi/mos/certs/ca-sas.crt
topic # both 0
```

The configuration files for brokers 2 and 3 would look similar to the one below.

```
port 1883
log_type all
listener 8883
psk_hint my test bridge
psk_file c:\mos\certs\psk_file.txt
#cafile c:\mos\casas\ca-sas.crt
#keyfile c:\mos\casas\sas-server.key
#certfile c:\mos\casas\sas-server.crt
```

Testing The Connections

When you connect the bridge there is actually no indication that a secure connection is being used provided that the configuration is OK.

However you will get an indication if you have a configuration problem.

The screen shot below show the connection problem that I caused by using a **mismatched key** for connection **bridge-02**.

```
C:\mos>mosquitto -c bridge.conf
1519324828: mosquitto version 1.4.14 (build date 2017-07-10 23:55:18+0100) start
ing
1519324828: Config loaded from bridge.conf.
1519324828: Opening ipv6 listen socket on port 1883.
1519324828: Opening ipv4 listen socket on port 1883.
1519324828: Opening ipv6 listen socket on port 8883.
1519324828: Opening ipv4 listen socket on port 8883.
1519324828: Bridge local.ws6.bridge-02 doing local SUBSCRIBE on topic #
1519324828: Connecting bridge bridge-02 (sas:8883)
1519324828: Bridge ws6.bridge-02 sending CONNECT
1519324828: OpenSSL Error: error:140943FC:SSL routines:ssl3_read_bytes:sslv3 ale
rt bad record mac
1519324828: OpenSSL Error: error:1409E0E5:SSL routines:ssl3_write_bytes:ssl hand
shake failure
1519324828: Socket error on client local.ws6.bridge-02, disconnecting.
```

Video- How to Create a Secure Bridged Connection on Mosquitto

Questions and Answers

Q- What is the PSK Hint?

A- See this stackoverflow response

Q- Is PSK less secure than using a certificate?

A- Probably yes but opinions seem to vary- see here. PSK is however much easier to implement than certificates.

Was This Useful?

References:

- RFC 4279-

Related Tutorials

- Mosquitto MQTT Bridge-Usage and Configuration
- Mosquitto SSL Configuration -MQTT TLS Security
- SSL and SSL Certificates Explained
- MQTT Security Mechanisms

Please rate? And use Comments to let me know more

43 comments

John says:

July 12, 2022 at 8:01 pm

Hey Steve,

Do any of the Paho clients (java, c, etc.) support PSK?

Reply

steve says:

July 13, 2022 at 3:27 pm

Hi

Not aware of any and I find it strange as it is an easier option than certificates.

I did find this on a quick search

https://github.com/eclipse/paho.mqtt.python/issues/451

I'm reluctant to use anything that is not standard as it might not work going forward.

Rgds

Steve

Reply

Jakob says:

March 31, 2022 at 12:45 pm

Hi.

Thanks for everything, Steve. Your site is pure gold.

Just a quick comment. As far as I can tell from you need to ad the following in order to get PSK to work on bridges when it comes to newer versions of Mosquitto:

use_identity_as_username true

Otherwise the bridge_identity is ignored server-side.

So you would have for instance:

listener 8883

psk_hint bridge

psk_file /mosquitto/psk_file.txt

use_identity_as_username true

BR

Jakob

Reply

steve says:

March 31, 2022 at 8:01 pm

Hi

Tks for taking the time to point that out I will add it to the tutorial

rgds

steve

Reply

Manuel says:

January 6, 2022 at 7:34 pm

Hi Steve

Thank you very much for your explanations.

Is it possible to configure two bridges, each one with different SSL settings?

Thanks in advance.

Manuel

Reply

---

steve says:

January 7, 2022 at 9:43 am

Hi

Is that two bridges on the same broker? If so I haven't tried it but will give it a try if that is the question.

What settings do you mean?

rgds

steve

Reply

Manuel says:

January 7, 2022 at 11:37 am

Hi Steve,

Thank you very much for the quick reply.

I have the following configurations, but it seems to do not work. If I configure only one bridge it works, but with two bridges it fails. Maybe it is not related to the configuration but something else.

Can I use different certificates for each configured bridge?

Another question, if I configure for instance, bridge-host1 on other port, ex:443, which port shall I configure as a listener

Thanks in advance,

Manuel

```
### Bridge Config ###

log_type all
tls_version tlsv1.2

listener 8884

connection bridge-host1
address 192.168.10.100:8883
#remote_username broker-01
#remote_password 1234
topic # out 0 "" ""
topic # in 0 "" ""
cafile /udata/config/certs/ca_host1.crt
certfile /udata/config/certs/cserver_host1.crt
keyfile /udata/config/certs/server_host1.key

connection bridge-host2
address 192.168.10.200:8883
#remote_username broker-02
#remote_password 5678
topic # out 0 "" ""
topic # in 0 "" ""
cafile /udata/config/certs/ca_host2.crt
certfile /udata/config/certs/cserver_host2.crt
keyfile /udata/config/certs/server_host2.key

##################################################

### broker-01 Conf ###
tls_version tlsv1.2
listener 8883
cafile /udata/config/certs/ca-host1.crt
certfile /udata/config/certs/server_host1.crt
keyfile /udata/config/certs/server_host1.key
allow_anonymous false
password_file /customer_apps/password.txt

##################################################

### broker-02 Conf ###
tls_version tlsv1.2
listener 8883
cafile /udata/config/certs/ca-host2.crt
certfile /udata/config/certs/server_host2.crt
keyfile /udata/config/certs/server_host2.key
allow_anonymous false
password_file /customer_apps/password.txt
```

Reply

Manuel says:

January 7, 2022 at 11:41 am

Hi Steve,

Thank you very much for the quick reply.

I have the following configurations, but it seems to do not work. If I configure only one bridge it works, but with two bridges it fails. Maybe it is not related to the configuration but something else.

Can I use different certificates for each configured bridge?

Is it possible to use certificates protected with password? If so, how ca I do it?

Another question, if I configure for instance, bridge-host1 on other port, ex:443, which port shall I configure as a listener

Thanks in advance,

Manuel

```
### Bridge Config ###

log_type all
tls_version tlsv1.2

listener 8884

connection bridge-host1
address 192.168.10.100:8883
#remote_username broker-01
#remote_password 1234
topic # out 0 “” “”
topic # in 0 “” “”
cafile /udata/config/certs/ca_host1.crt
certfile /udata/config/certs/cserver_host1.crt
keyfile /udata/config/certs/server_host1.key

connection bridge-host2
address 192.168.10.200:8883
#remote_username broker-02
#remote_password 5678
topic # out 0 “” “”
topic # in 0 “” “”
cafile /udata/config/certs/ca_host2.crt
certfile /udata/config/certs/cserver_host2.crt
keyfile /udata/config/certs/server_host2.key

################################################

### broker-01 Conf ###
tls_version tlsv1.2
listener 8883
cafile /udata/config/certs/ca-host1.crt
```

certfile /udata/config/certs/server_host1.crt
keyfile /udata/config/certs/server_host1.key
allow_anonymous false
password_file /customer_apps/password.txt

##

broker-02 Conf
tls_version tlsv1.2
listener 8883
cafile /udata/config/certs/ca-host2.crt
certfile /udata/config/certs/server_host2.crt
keyfile /udata/config/certs/server_host2.key
allow_anonymous false
password_file /customer_apps/password.txt

Reply

steve says:

January 7, 2022 at 12:10 pm

The files look ok. I would start simple which means two bridges no passwords and no SSL. Then add SSl and then passwords.
You cannot password protect the certificate so if you did that then it will fail.
Rgds
Steve

Reply

Ragha says:

September 16, 2021 at 9:54 am

Hi Steve
There is some problem with topic name starting with $
when i removed the $ from the topic...bridge can receive as well send back the reply.
./mosquitto_pub -q 1 -t "$rps/registrations/res/202/?\$rid=1&retry-after=30" -m "{ \"status\": \"assigned\", \"operationId\": \"1\", \"registrationState\": { \"assignedURL\":\"ka.example.com\",\"deviceId\":\"40355b34-e7f5-4676-a1bc-33546254e1e1\",\"status\":\"assigned\"} }"
regards
Ragha

Reply

Ragha says:

September 15, 2021 at 4:50 pm

Hi Steve

Your knowledge store is of great help.

Related to bridge…my setup is as below

Machine1.MQTTClient(Pub)—>Over SSL–>Machine1.Broker—>SSL—>Machine2.Bridge —>Machine2.MQTTClient(Sub)

How to configure bridge for such a setup?

Reply

steve says:

September 15, 2021 at 6:29 pm

the brokers would need to use a different port for the bridge. If we assume standard 1883 for clients.

broker 1 listens on 1883 with SSL.

Broker 1 is bridge to broker 2 using port 1884

broker 2 is listening on 1884 with SSL.(bridge)

broker 2 listening on 1883 for client connections

client2 connects to broker 2 on 1883

Haven't tried it but if you have probs let me know and I will try it.Also let me know if it works.

Rgds

Steve

Reply

Ragha says:

September 16, 2021 at 9:59 am

Thanks

Broker config file

listener 1883

protocol mqtt

allow_anonymous true

socket_domain ipv4

log_type all

connection_messages true

connection kaazing.example.com

#address 192.168.56.103:9001

address kaazing.example.com:8883

bridge_cafile /root/myCA.pem

#bridge_protocol websocket

topic # out 1

topic # in 1

topic # out 0

topic # in 0

My broker config file:

listener 1883

protocol mqtt

allow_anonymous true

listener 8883

protocol mqtt

socket_domain ipv4

log_type all

keyfile /root/ragha/mosquitto/git/localDocker/server.key

certfile /root/ragha/mosquitto/git/localDocker/server-crt.signedby.myca.pem

connection_messages true

With removal of $ from the topic name….things hv started to work

Reply

---

rucksman007 says:

May 24, 2021 at 12:05 pm

Hi Steve,

first of all thanks for your excellent tutorial. I have used them a lot, especially for Node Red.

Today I have a question regarding this tutorial. My use case is as follows;
– Internal Server with Node Red and Mosquitto
– Internal Server is not accessible from outside

Now I want to be able to use MQTT also from outside of my home. Of course I could open my firewall, do a port forwarding and use some DynDNS service to reach my internal server, but this is not what I want to do. So my idea is to run a second broker on some small external VPS. When not at home I use this broker, when at home I use the broker at home. This requires both brokers to be bridged, and I want to secure this connection with encryption.

So my goal is:
– All my IoT devices at home talk to the internal broker (not encrypted)
– When on the road my smartphone talks to the external broker which sends and receives the messages to/from the internal broker (encrypted)

Is this something achievable? If yes, whould woud be the configuration? Thanks for helping

Reply

---

steve says:

May 24, 2021 at 6:23 pm

Yes

The local broker would be the bridge and the external broker would require SSL. It is important that the local broker does the bridge as it is establishing the connection and so doesn't require port forwarding.

Does that make sense?

Rgds

Steve

Reply

rucksman007 says:

May 25, 2021 at 6:56 am

Hi Steve,

thanks for your answer. I ordered a small VPS yesterday and will try to do the setup as per your recommendation. If I encounter problems, I will probably ask more questions here. Thank you!

Reply

rucksman007 says:

June 12, 2021 at 2:29 pm

Hi Steve,

finally I found the time to configure my broker und the server.

Broker and server do communicate well until I configure SSL/TLS. Without TLS I publish a message to the external broker (which is the server) and this message is immediately recieved by the internal broker (the bridge). When using SSL/TLS, no message is received from the bridge. I can connect to the external broker with MQTT Explorer from my windows machine without any problem, but when I publish a message, nothing is received by the local broker. The log from the external broker says:
"1623507661: OpenSSL Error[0]: error:14094418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca"
and from time to time also
"1623507854: OpenSSL Error[0]: error:1408F10B:SSL routines:ssl3_get_record:wrong version number"

Any idea where I should start searching for the problem? I used exactly the commands from you guide to mosquitto TLS.

Reply

steve says:

June 12, 2021 at 3:23 pm

The way it should be setup is
local broker is the bridge.
External server requires SSL for the bridge connection
for clients connecting to the external broker they can usually connect using

plain mqtt and I would configure that to start even though you might want to change it later.
This means that the CA would be created for external server and copied to the local server.
Is that how you set it up?
Rgds
Steve

Reply

rucksman007 says:

June 12, 2021 at 3:31 pm

Sorry for my post regarding problems with SSL/TLS. All of my problems were answered here already.
Problem 1: "wrong version number" was solved with "tls_version" entry in server conf
Problem 2: "unknow ca" and a not reported error message on the internal broker: I used the exactly same credentials for ca.crt and server.csr which leads to problems. In a second attempt I changed "Organization Name" in server.csr to a different value from ca.crt, and from then on no errors and enryption between external and internal broker works fine (at least I hope so, testing this is not that easy as you already wrote). Thanks for your wonderful tutorials!

Reply

John Ronan says:

February 17, 2021 at 6:18 pm

Hi Steve,
I used your instructions today to get a bridge working. However in the absence of a username/password, "allow_anonymous true" is needed on broker2 for your example above to work.
Cheers
John

Reply

steve says:

February 17, 2021 at 7:12 pm

John
I assume that the other broker is running v2. In v2 the security settings for the install changed so it defaults to allow anonymous false.
I will update the tutorial to mention it.
Rgds
Steve

Reply

fernando says:

December 2, 2020 at 11:36 am

Hi, thanks a lot for your tutorial. I am a newbee and i have a question about certificates and let's encrypt. In CLOUD, I have certbot to renew my certificates from let's encrypt. These "files" are in the cloud.

Now, i have a broker in local which I want to act as bridge... (i will send command to it from my tasmotas devices). but where could i get that ca file? and will I have to update that file every time certbot renews my certificates files in my cloud server?

Thanks.

Reply

steve says:

December 2, 2020 at 5:30 pm

You should be able to download the CA file and you would need to renew them but I don't thinks it is checked on the broker anyway. What broker are you using in the cloud? I would opt for a pre shared key if possible it is easier and just as secure when set up.
Rgds
Steve

Reply

fernando says:

December 2, 2020 at 11:31 pm

Hi Steve, thanks a lot. I am using google cloud virtual machine with MOSQUITTO broker installed (+nodered +grafana +...)
In my local openwrt router, I have also mosquitto broker installed.
Virtual machine is certificated by Lets Encrypt with its usual cronbot job.
I am trying with Certificate encryption in order to understand and learn.
I dont understand very well about certificates. I get managed to certificate my google cloud virtual machine, open ports, get mosquitto with SSL working, also nodered ui, ...
But with bridge, I dont understand exactly where I could find my CA file to download to my local router. And also, if that file also change every 60-90days, or it is a file from let's encrypt as it is a CA authorithy, isn't it? So it will no change in "years".
Thanks for your help

Reply

steve says:

December 3, 2020 at 1:02 pm

Hi

You need the CA from lets encrypt the other files like server.key are held in the cloud and will need updating. The CA should be a long period.

Take a look here

https://community.letsencrypt.org/t/where-can-i-download-the-trusted-root-ca-certificates-for-lets-encrypt/33241

Let me know if that helps.

Rgds

Steve

Reply

Fernando says:

December 3, 2020 at 9:43 pm

Hi, I have finally solve with “opkg install ca-certificates” and with this line in config file: “bridge_capath /etc/ssl/certs/”

Thanks a lot for your help, bridge is working with SSL in port 8883.

steve says:

December 5, 2020 at 3:49 pm

Hi

I cam across this article which you should find useful if you haven’t seen it

https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-the-mosquitto-mqtt-messaging-broker-on-ubuntu-18-04-quickstart

fernando says:

December 2, 2020 at 11:32 pm

Please, you say: “You should be able to download the CA file and you would need to renew them but I don’t thinks it is checked on the broker anyway”. Are you meaning that that file is in my google cloud virtual machine?

Thanks.

Reply

geek says:

September 25, 2020 at 3:20 pm

Your MQTT videos and posts are really great Steve.. !Thank you for such wonderful content…

I tried to connect from the local system to the virtual machine in the cloud using bridge concept. I have the TLS encryption in the local system broker and i have included the configuration settings for bridge in the VM broker as ,

External MQTT Broker
connection vm
address
bridge_cafile /etc/mosquitto/certs/ca.crts
topic # in
topic # out

But i cant send the data over the bridge. Is it because of my personal local pc IP is not accessible in the cloud VM? or is there any mistake i am making?
while running commands in the terminal, i am opening two terminals in the local pc for one sub and one pub. Then which ip need to be give there? In the cloud vm part also , in the terminal which ip i have to mention?
Please reply . I am stuck with this.

Reply

steve says:

September 25, 2020 at 6:24 pm

Hi
You will need to setup port forwarding on your home router see here
https://stevessmarthomeguide.com/understanding-port-forwarding/

Open up the 1883 and 8883 ports
Also I would try first without tls and then with tls
rgds
steve

Reply

geek says:

September 26, 2020 at 4:23 pm

Is there any other method instead of bridge concept to send data directly from device to cloud using mqtt? Also, one more doubt. Is it because of this local ip issue that I am unable to connect ? I have also tried to set up the bridge with two virtual machine instance in the cloud as a testing purpose. But its also not working.Data is not being published in the instance terminal after giving sub in the first instance. Looking forward to hear from you.
Thanks and regards,
Geek.

Reply

steve says:

September 26, 2020 at 4:45 pm

Yes you can send directly from the client to a cloud broker. That way you don’t need any router setup.

Rgds

Steve

Reply

geek says:

September 26, 2020 at 6:11 pm

Thank you for sparing your time.

Are you speaking about cloud pub sub? I am new to this cloud and mqtt so please clear this.

I sent data with google api and pub sub. But i want mqtt broker in my system to connect to cloud. That is what i am trying to do. Do it need bridge ?

Thanks and regards,

Geek

steve says:

September 26, 2020 at 7:20 pm

There are a number of cloud based brokers you can use.

Brokers like

test.mosquitto.org

are for testing and not production

other like cloudmqtt.com are for production. Cloudmqtt don't any longer provide a free plan but they did when I wrote this tutorial.

http://www.steves-internet-guide.com/create-mqtt-broker-cloudmqtt/

However beebotte.com does so you could try that.

There is a list of public broker in this tutorial

http://www.steves-internet-guide.com/mqtt-hosting-brokers-and-servers/

Sebastian says:

April 28, 2020 at 3:36 pm

Hello Steve,

I am desperately trying to get my setup up and running. Here is what I want to achieve:

I have a webhosted vServer running the main broker; it is configured to require a certificate.

Then I have three RPi zero hosted bridges from my home, office and workshop, each with its own client certificate. Those bridges receive unencrypted data from several devices, mostly tasmota.

Now here's the funny thing: If I run mosquitto „manually" as user pi, it connects successfully, sending up and down. The main service, probably run as root, fails. Can

you please do a post about users and permissions? Do I need to generate the certificates on the machine where they are actually used?

Best wishes and stay safe in these hard times.

Sebastian

Reply

steve says:

April 28, 2020 at 5:18 pm

Hi
Mosquitto runs as the user mosquitto and doesn't require root permissions.
I assume you are talking about the central broker> I would check that mosquitto user has permissions to read the conf and ca files as it is probably a permission issue.
Rgds
Steve

Reply

vikas says:

October 1, 2019 at 8:16 am

your articals are very very good and it helps me for upgrading my skills but there is so many many problems when implementing so it is best to provide step by step implementation on pc and solve many problems occurs it will help a lot
thanlksss

Reply

Lukas says:

February 18, 2019 at 5:20 pm

GREAT Guide Thanks,

for anyone struggling with old Mosquitto versions and bridges:
eg. in 1.3 Mosquitto the remote_username and other params where not using the remote_* prefix

See here
https://mosquitto.org/ChangeLog.txt

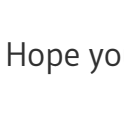
Reply

DFarrell says:

December 3, 2018 at 4:50 pm

For a bridge connection using SSL I added a bridge.conf file using capath. I have Let's Encypt as a CA.
connection bridge-01
address myhost.com:8883
bridge_capath /etc/ssl/certs/
remote_username mqttu
remote_password mqttpw
topic home/# out 0

Reply



Mauricio says:

July 17, 2018 at 12:56 am

Hi, you have great tutorials for Mosquitto. Your tutorials have help me to configure my brokers in the right way. Thank you so much.

I have a problem with this tutorial, I already make a connection between two mosquitto brokers without encryption and it works well. I had some troubles making one of the brokers to use encryption but I finally get it to work using your tutorial. I can connect to that broker with a client using the ca.crt file and it works well. But I can't make a bridge between my two brokers using the ca.crt file.

The log is showing an error every time it tries to connect:
OpenSSL Error: error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number.

Is there a problem whith my tls version from the config file? Is there a problem if one of my brokers is local and the other one has a domain?

Hope you can help me out.

Thanks.

Reply

steve says:

July 17, 2018 at 8:35 am

Looks like a problem with the .conf file. Can you send me the .conf file for the client side of the bridge or both if you are not sure.
Use the ask-steve page on the site to send it

Reply

Robert says:

May 16, 2020 at 3:48 pm

Hello, thanks for youre awesome content Steve!

Im using Node-red to send random values across a bridge and subscribe to them on the other side but getting an error.

Error: OpenSSL Error: error:xxxxx:SSL routines:SSL3_GET_RECORD:wrong version number

It appears I am getting the same error as others but have narrowed it down to the 'local' MQTT subscribe nodes attempting to subscribe to data from broker 2. Im suspecting the error arises as the local subscribe node doesnt share the same PSK/TLS. My setup on two node-red instances is:

local MQTT pub node ——-> Broker 1 Broker 2 ——> local MQTT sub node

Do you know of any way for the local MQTT subscribe node to subscribe to the bridged broker without needing TLS to subscribe to broker 2? Assuming that is the problem..

Config:
Broker 1

listener 1883
listener 8883
log_type all
connection bridge-01
address 192.168.2.100:8883
bridge_identity bridge1
bridge_psk dd230d622
topic # both 0

Broker 2:

port 1883
listener 8883
log_type all
psk_hint initiate
psk_file /mosquitto/psk.txt

Cheers,
Robert

Reply

steve says:
May 16, 2020 at 6:07 pm

Hi
From what I can see you aren't using ssl but a preshared key for the bridge.

Copyright © 2011-2024 Steve's internet Guide
By Steve Cope

Does that make sense

Rgds

Steve

Reply

Leave a Reply

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

Comment *

Name *

Email *

Website

Post Comment