How do you set up encrypted mosquitto broker like a webpage which has https?

I'm trying to setup a mosquitto broker which is encrypted using ssl/tls. I don't want to generate client certificates. I just want an encrypted connection.

The man page only described the settings which are available, not which are needed and how they are used.

Which settings are needed and how do you set them?

I use mosquitto 1.3.5

ssl encryption certificate mqtt mosquitto

asked Oct 30 '14 at 15:39

Gussoh
251 1 2 9

2 Answers

There is a small guide here, but it does not say much: http://mosquitto.org/man/mosquitto-tls-7.html

You need to set these: certfile keyfile cafile

They can be generated with the commands in the link above. But easier is to use this script: https://github.com/owntracks/tools/blob/master/TLS/generate-CA.sh

After running the script and changing the config it could look something like this:

```
listener 8883
cafile /etc/mosquitto/certs/ca.crt
certfile /etc/mosquitto/certs/hostname.localdomain.crt
keyfile /etc/mosquitto/certs/hostname.localdomain.key
```

If mosquitto says `Unable to load server key file` it means that the user which is running mosquitto does not have permission to read the file. Even if you start it as root the broker might start as another user, mosquitto for example. To solve this do e.g. `chown mosquitto:root keyfile`

To connect to the broker the client will need the ca.crt-file. If you do not supply this the broker will say something like:

> OpenSSL Error: error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number

To supply it to the mosquitto_sub command you use `--cafile pathToCaCrt` . The ca.crt can be distributed with the clients and it will make sure that the server it is connected to actually is the correct server.

The `--insecure` flag of mosquitto_sub does not make the client accept all certificates (like with wget or similar), it just allows that the certificate not to have the host you are connecting to in common name. So you should make sure your certificate has your broker host as common name.

edited Jun 26 '15 at 8:03

ralight
5,226 2 20 41

answered Oct 30 '14 at 15:39

Gussoh
251 1 2 9

a handy guide here, that you can modify; jpmens.net/2013/09/01/installing-mosquitto-on-a-raspberry-pi – Matt. Oct 30 '14 at 22:46

To secure WebSocket access of Mosquitto, e.g. using a Let's Encrypt certificate, your config file could look like this:

```
listener 9001
protocol websockets
certfile /etc/letsencrypt/live/yourdomain.com/cert.pem
cafile /etc/letsencrypt/live/yourdomain.com/chain.pem
keyfile /etc/letsencrypt/live/yourdomain.com/privkey.pem
```

Make sure that the files are readable by Mosquitto (Debian in particular runs Mosquitto under the `mosquitto` user, which is unprivileged). You need Mosquitto 1.4 to support WebSockets.

To connect to this WebSocket using the Paho JavaScript client:

```
// host and port overwritten at connect
var mqtt = new Paho.MQTT.Client("yourdomain.com", 9001, "");

mqtt.connect({
    hosts: [ "wss://yourdomain.com:9001/" ],
    useSSL: true
});
```

Note that this does not imply any access control yet, so your MQTT broker will be publicly accessible. You may want to add authorization, too.

edited Aug 15 at 4:31

strugee
1,086 3 8 21

answered Dec 17 '15 at 11:54

romor
174 2 12

just wanna add on..if you are on AWS EC2, u will need to set the inbound rules of your security group to open the port (9001 in this case)...i got trolled real bad – Vic Aug 17 at 19:14