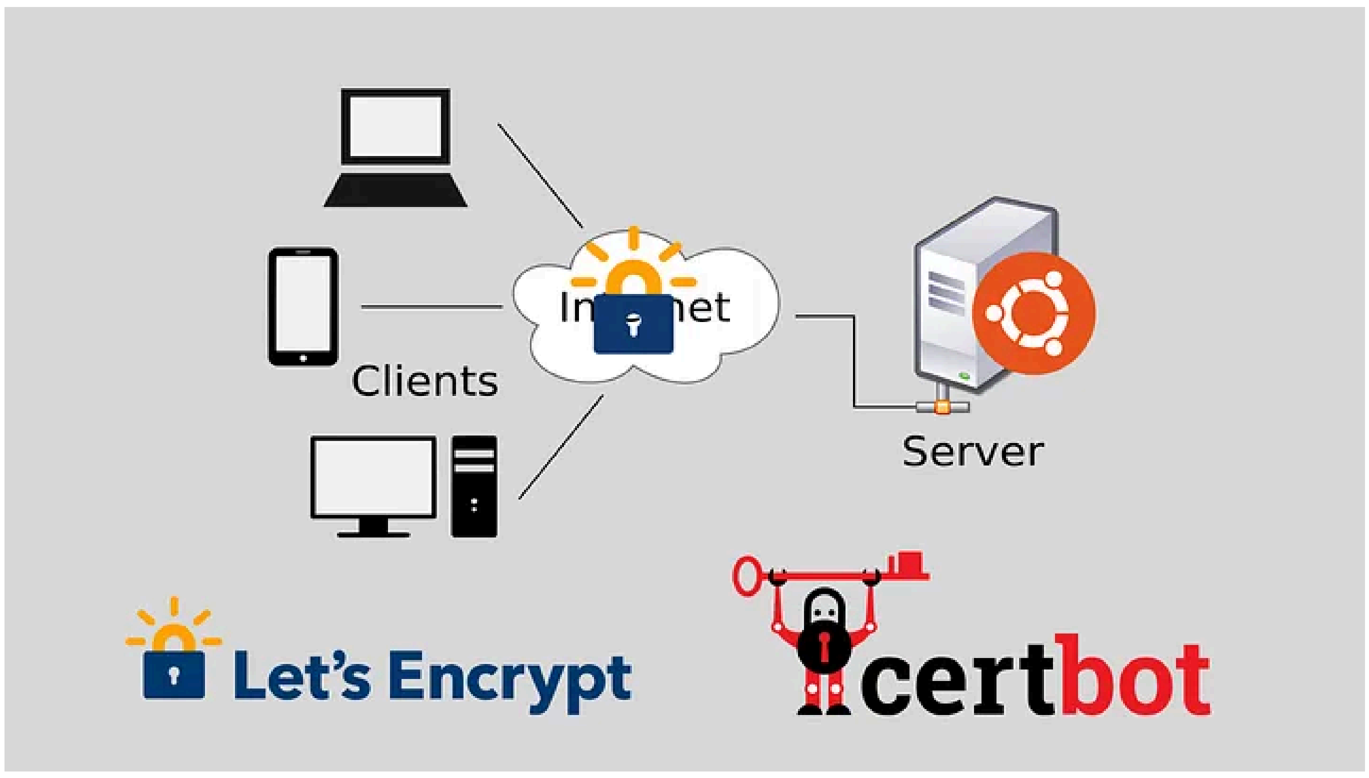
Enabling Transport Layer Security (TLS) for MQTT Mosquitto using Let's Encrypt and certbot

Besnik Belegu · Follow
7 min read · Sep 24, 2023

10 1



In my previous article (Setting up an MQTT Server — Part 1) I explained how you can setup a MQTT Mosquitto server but didn't go into details about using TLS for encrypting the communication from the device to the server.

In this article I will be using Let's Encrypt and Certbot to obtain a certificate and use it with your Mosquitto instance.

The setup will be done an on ubuntu server.

Prerequisites:

- You will need a domain name poiting to your server
- Mosquitto installed on your server (DUH!)
- Certbot installed on your server

I will assume you already linked the domain name to yor server, if you didn't, I will not go into that as it will be out of the scope of this article.

As for the setting up the Mosquitto server, if you haven't set up the server yet, you can find the details how to set it up here.

For Certbot, if you haven't installed it yet, you need to run the following command first:

```
sudo apt-get update && sudo apt-get install certbot
```

Once installed, you can now start the certificate obtaining process.

Since I already have something else running on my server, and port 80 and 443 are already used, I won't be able to do it the easy way which is by entering this command (replace <your_domain> with your actual domain):

```
sudo certbot certonly --standalone -d <your_domain>
```

If I run it like this on my server I will get a message saying:

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for your.domain.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Could not bind TCP port 80 because it is already in use by another process on
this system (such as a web server). Please stop the program in question and then
try again.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(R)etry/(C)ancel:
```

If you are facing the same issue as I did, you need to press **C** to cancel this process and proceed with the next steps. Otherwise, you can follow the prompts to complete the certificate request.

Next step is to use a DNS Plugin wich will allow you to prove the domain ownership by adding a DNS TXT record.

To do this you need to write the following command:

```
sudo certbot certonly --manual --preferred-challenges dns -d your_domain
```

Now follow the prompts to add the required DNS TXT record to your domain's DNS settings. It will also explain to you how you can test if the DNS TXT record is added successfully.

If all went well (as it should), you will now have the certificates somewhere like *'/etc/letsencrypt/live/your.domain.com/'.*

There should be two new files created there, *fullchain.pem* and *privkey.pem.*

For Mosquitto you will need the *cafile*, *certfile* and the *keyfile*.

You will need the *certfile* as well. *certfile* typically contains only the domain's certificate. Since *fullchain.pem* file contains your domain's certificate and the chain of Let's Encrypt intermediate certificates, you can extract the domain's certificate from *fullchain.pem* and save it to a *cert.pem.*

Extracting the Domain Certificate Manually

First you need to read the contents of *fullcain.pem.* You can do that by either opening the file with your prefered editor vim or nano, or just using cat to display it's contents on the terminal. For the purpose of this article I will use cat. Don't forget to replace *your.domain.com* with your actual domain.

```
sudo cat /etc/letsencrypt/live/your.domain.com/fullchain.pem
```

Since the file contains multiple certificates, each will be enclosed between ' — — -BEGIN CERTIFICATE — — -' and ' — — -END CERTIFICATE — — -' markers. The first one is usually your domain's certificate.

Copy the domain's certificate, including **BEGIN** and **END** markers to a new file, in our case *cert.pem*. I will do it using nano:

```
sudo nano /etc/letsencrypt/live/your.domain.com/cert.pem
```

Once the new file is opened, you can paste the contents of the copied certificate and save the file (with nano, you can do it by pressing CTRL+X then Y to confirm and Enter to exit).

Editing Mosquitto Configuration

Now that you have the certificates, you need to open the mosquitto.conf file which is located at */etc/mosquitto/mosquitto.conf*

```
sudo nano /etc/mosquitto/mosquitto.conf
```

Add the TLS configuration in the opened file by adding these lines before the line that reads **'include_dir /etc/mosquitto/conf.d'** (don't forget to replace your.domain.com with your actual domain).

```
listener 8883
cafile /etc/letsencrypt/live/your.domain.com/fullchain.pem
certfile /etc/letsencrypt/live/your.domain.com/cert.pem
keyfile /etc/letsencrypt/live/your.domain.com/privkey.pem
```

Close the file and save it.

To make sure the Mosquitto runs with the new configuration, you need to restart the server.

```
sudo service mosquitto restart
```

If all goes well, your Mosquitto instance will now be accepting TLS connections.

Make sure when you connect you specify in the client application that you are using TLS and that the server port is 8883.

To check the logs while the Mosquitto is running you can use tail to monitor the log like this:

```
sudo tail -f /var/log/mosquitto/mosquitto.log
```

Note

The Let's Encrypt Certificates are valid for 90 days from the day you obtain them. The above mentioned process you used will not renew the *cert.pem* file you created manually. Therefor, whenever you renew the ***fullchain.pem*** and ***privkey.pem*** files, you will need to repeat the manual process of creating the ***cert.pem*** file again, or you can automate it with a script, which is out of the scope of this article.

. . .

Troubleshooting

When I did this process, I ran into some permissions issues where the Mosquitto didn't have permissions to acces the .pem files inside ***/etc/letsencrypt/live/your.domain.com/***

I solved this by first copying the certificates to ***/etc/mosquitto/certs*** like this:

```
sudo cp /etc/letsencrypt/archive/your.domain.com/fullchain1.pem /etc/mosquitto/c
sudo cp /etc/letsencrypt/archive/your.domain.com/privkey1.pem /etc/mosquitto/cer
sudo cp /etc/letsencrypt/live/your.domain.com/cert.pem /etc/mosquitto/certs/cert
```

As you will notice the place where I'm copying the files is ***/etc/letsencrypt/archive/…*** and not ***/etc/letsencrypt/live/…*** since the files in live folder are symbolic links to the actual files. The *cert.pem*, since we created it manually, will be on the live folder.

Once you've copied the files You need to ensure that Mosquitto user has read and execute permissions to the copied files by running the following commands:

```
sudo chown mosquitto:mosquitto /etc/mosquitto/certs/fullchain.pem /etc/mosquitto
sudo chmod 400 /etc/mosquitto/certs/fullchain.pem /etc/mosquitto/certs/privkey.p
sudo chmod +x /etc/mosquitto
sudo chmod +x /etc/mosquitto/certs
```

You can verify the permissions like this:

```
sudo -u mosquitto ls /etc/mosquitto/certs
```

If the correct permissions are set, you should be able to see the contents of the ***/etc/mosquitto/certs*** directory without any issues.

Now that you have change the location of the certificates, you also need to make the changes in ***/etc/mosquitto/mosquitto.conf*** file to point to the correct path where the certificates are located.

Now restart the Mosquitto service and you should be up and running.

```
sudo service mosquitto restart
```

. . .

Best Practices for Enabling TLS for Mosquitto with Let’s Encrypt and Certbot

When diving into the world of securing Mosquitto with TLS, it’s essential to keep a few best practices in mind. Let’s explore some of them.

Regular Certificate Renewal: One of the quirks of Let’s Encrypt certificates is their 90-day validity. It’s a good idea to either set up automatic renewals or, at the very least, have calendar reminders. This way, you won’t face any unexpected service interruptions due to expired certificates.

Backup, Backup, Backup: Always have backups of your certificates and keys. It’s like having an insurance policy for your server. If anything goes awry, you’ll be thankful for that secure backup you stashed away.

Guard Your Private Keys: Treat your `privkey.pem` like a treasure. It should be accessible only to those who absolutely need it. Tightening its permissions ensures it remains confidential and out of the wrong hands.

Opt for Strong Cipher Suites: While the defaults with Mosquitto and Certbot are pretty solid, it doesn’t hurt to periodically review the cipher suites in use. The world of cybersecurity is ever-evolving, and staying updated ensures you’re always on top of your game.

Keep an Eye on Certificate Expiry: Consider using monitoring tools that can give you a nudge when your certificates are about to expire. It’s always better to be proactive than reactive.

Stay Updated: Just like any software, Mosquitto and Certbot receive updates. Regularly updating them ensures you’re shielded with the latest security patches and features.

Dedicate a User for Mosquitto: It’s a wise move to run Mosquitto under its own dedicated user. This user should have just enough privileges to do its job and nothing more. It’s all about minimizing potential risks.

Firewall Savviness: If you’ve transitioned to MQTT over TLS and aren’t looking back, think about closing off port 1883. It’s all about keeping things tight and secure.

Test, Test, and Test Again: Whenever you renew or tweak your certificates, give your MQTT connections a test run. It's a simple step that can save you from future headaches.

Stay in the Loop: Lastly, always keep an ear to the ground. Security best practices aren't set in stone. They evolve. Regularly checking for updates and new recommendations ensures your Mosquitto setup remains both modern and secure.

. . .

Wrapping Up

Securing our Mosquitto broker with TLS using Let's Encrypt and Certbot is more than just a technical exercise; it's a commitment to ensuring the safety and integrity of our data and communications. As we've journeyed through the process and delved into best practices, it's evident that while tools and configurations play a crucial role, the real essence lies in our approach to security. Being proactive, staying informed, and regularly revisiting our setups can make all the difference. As the world of IoT continues to expand and the importance of MQTT grows, let's pledge to make security not just a task, but a habit. Here's to safer, more reliable MQTT communications!

Well, in my opinion, absolute security is an illusion. In the real world, security is about managing and mitigating risks rather than eliminating them entirely. It's a continuous journey of adaptation, learning, and vigilance. And to echo the words of Kevin Mitnick, one of the most renowned computer security experts:

> "The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards."

Mqtt Lets Encrypt Certbot Mosquitto IoT

Written by Besnik Belegu

5 Followers · 13 Following

Follow

I like to read, sometimes write but mostly try out new things. Hope you enjoy what I write and find it useful.

Responses (1)

What are your thoughts?

Respond

Stephen Nelson
Jan 2, 2024

I'm setting up the system, and I've run into the file permissions you mention in the "Troubleshooting" section. Copying the files would work fine temporarily, but how did you solve the problem in a way that would survive a renewal?

1 reply Reply

More from Besnik Belegu

Besnik Belegu

Setting up an MQTT Server

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging...

Sep 23, 2023 17

Besnik Belegu

Enabling and Securing MQTT WebSockets with TLS

In our journey to set up a robust MQTT server, we've previously discussed setting up an...

Sep 25, 2023 2

Besnik Belegu

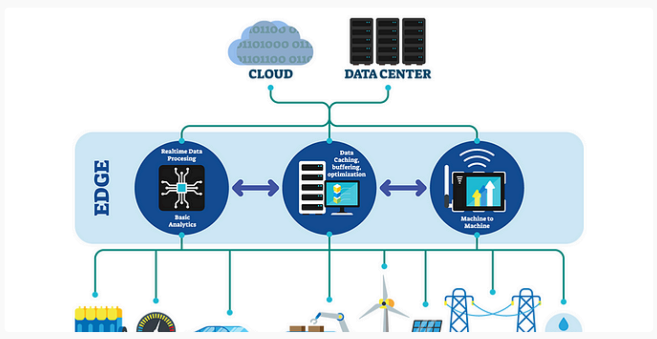
A Little Tool I Made: Satellite Device Phonebook Manager

In this article, I share my journey of creating a Satellite Device Phonebook Manager using...

Sep 26, 2023

See all from Besnik Belegu

Recommended from Medium



In Modernizing Maintenance by Stylianos Chiotis

MQTT or OPC UA | My Roadmap to IIoT Communication Protocols

Through the eyes of a Mechanical Maintenance Engineer

Aug 19, 2024 1

In AI Innovator From PrismAI by Abhijat Sarari

License Plate Recognition with OpenCV and Tesseract OCR

License Plate Recognition (LPR) is a powerful tool in computer vision, used in applications...

Nov 4, 2024 53

Lists

Staff picks
800 stories · 1569 saves

Always Free
24 GB RAM + 4 CPU + 200 GB
FREE
@harendraverma2 @harendra21 @harendra21

Harendra

How I Am Using a Lifetime 100% Free Server

Get a server with 24 GB RAM + 4 CPU + 200 GB Storage + Always Free

Oct 26, 2024 8.5K 132

Subrahmanya Gonella

Mastering GitLab CI/CD on Proxmox: Build Faster, Test...

Building on the foundation of GitLab on Proxmox, this guide dives into dedicated...

Jan 3 4

In Stackademic by Crafting-Code

I Stopped Using Kubernetes. Our DevOps Team Is Happier Than Ever

Why Letting Go of Kubernetes Worked for Us

Nov 19, 2024 4.91K 145

In Level Up Coding by Jacob Bennett

The 5 paid subscriptions I actually use in 2025 as a Staff Software…

Tools I use that are cheaper than Netflix

Jan 7 5.2K 115

See more recommendations

Help Status About Careers Press Blog Privacy Terms Text to speech Teams