Technical Report

Masudul Hasan Masud Bhuiyan

Overview:

The goal of this project is to create a self signed certificate for a web server and embed that certificate to the trusted CA list so that browsers can identify the certificate as a valid one.

Current Status:

Currently this project can successfully create self signed certificate for a website and can embed that that certificate locally. But it ask for the root permission to run the script.

Work Flow:

There are 3 steps in to run this project:

1. configure a server in local machine (Apache)
2. create a self signed certificate for the **localhost/127.0.0.1**
3. Add the created certificate to trusted CAS list to make it a valid certificate

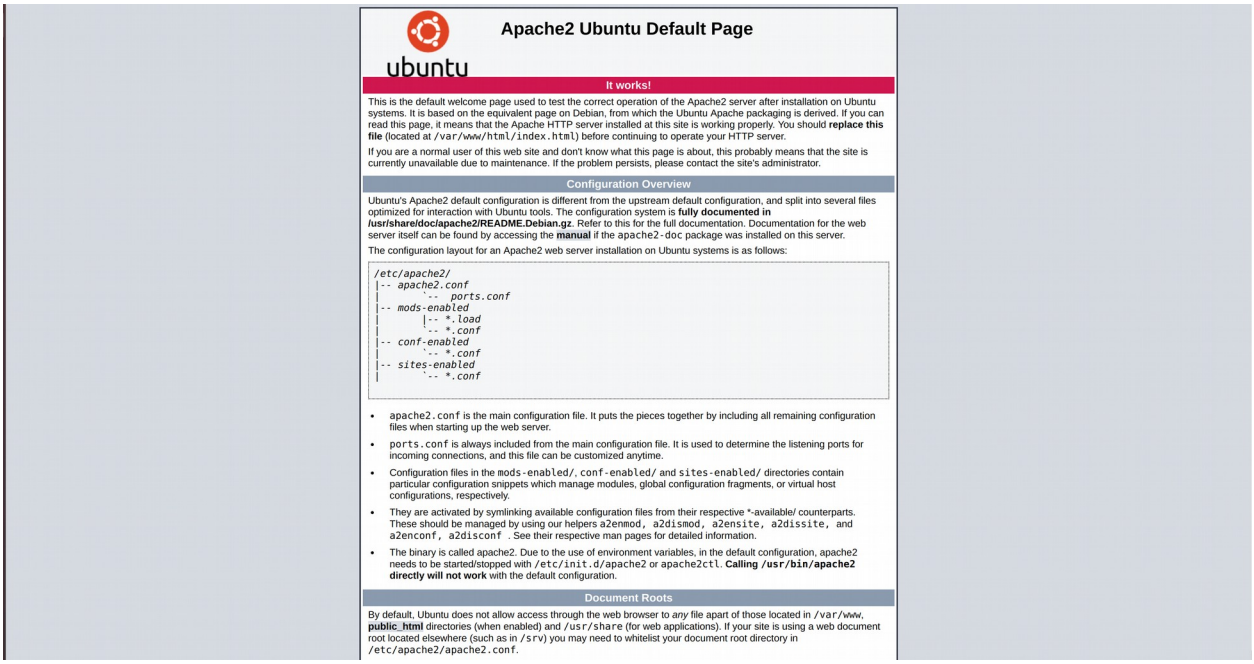
Step 1:

The first step is to configure a web server in which our website will be hosted and it will provide the certificate for our website to the browser. I have used **Apache-2** as my web server. I configured apache server in my local machine. I am using Ubuntu 18.04.02 as my local machine.

This link has step by step instruction for installing and configuring apache server in ubuntu.

https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04

If the installation is correct it show this page:



step 2:

Now we have to create self signed certificate for out webpage/ server. To create a self signed certificate we have to create a public key. I automated this instruction in **"create_self_certificate.py"** to create and put the necessary files in the right directory. To make the other necessary changes for the server **"make_configuration.sh"** files is used. Both files need root/sudo permission to run.

"create_self_certificate.py" will create key and certificate named “apache-selfsigned.key” and “apache-selfsigned.crt”. It will copy these file to *"/etc/ssl/private/apache-selfsigned.key"* and *"/etc/ssl/certs/apache-selfsigned.crt”* respectively.

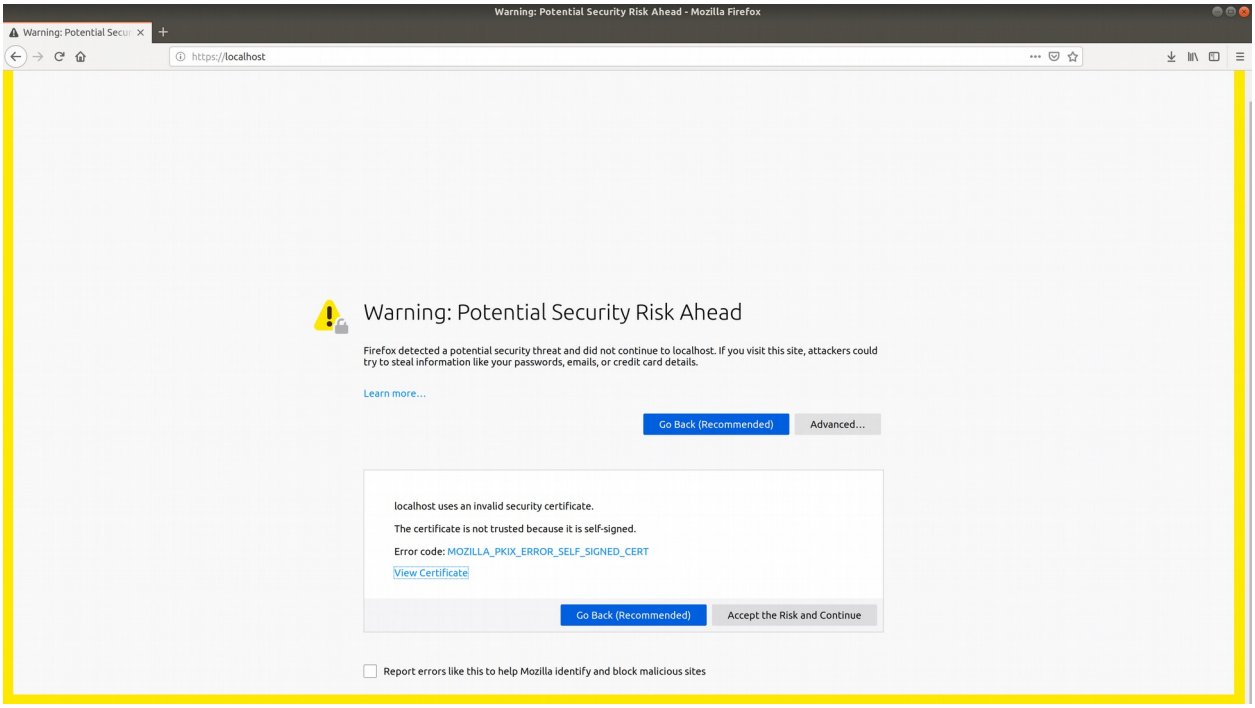
It will also copy other necessary configuration file for the server to appropriate directory.

"make_configuration.sh" setup the necessary environment for the server and restart the server so that it can provide secure connection.

This article gives step by step instruction on how to create self signed certificate.
https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-18-04

step 3:

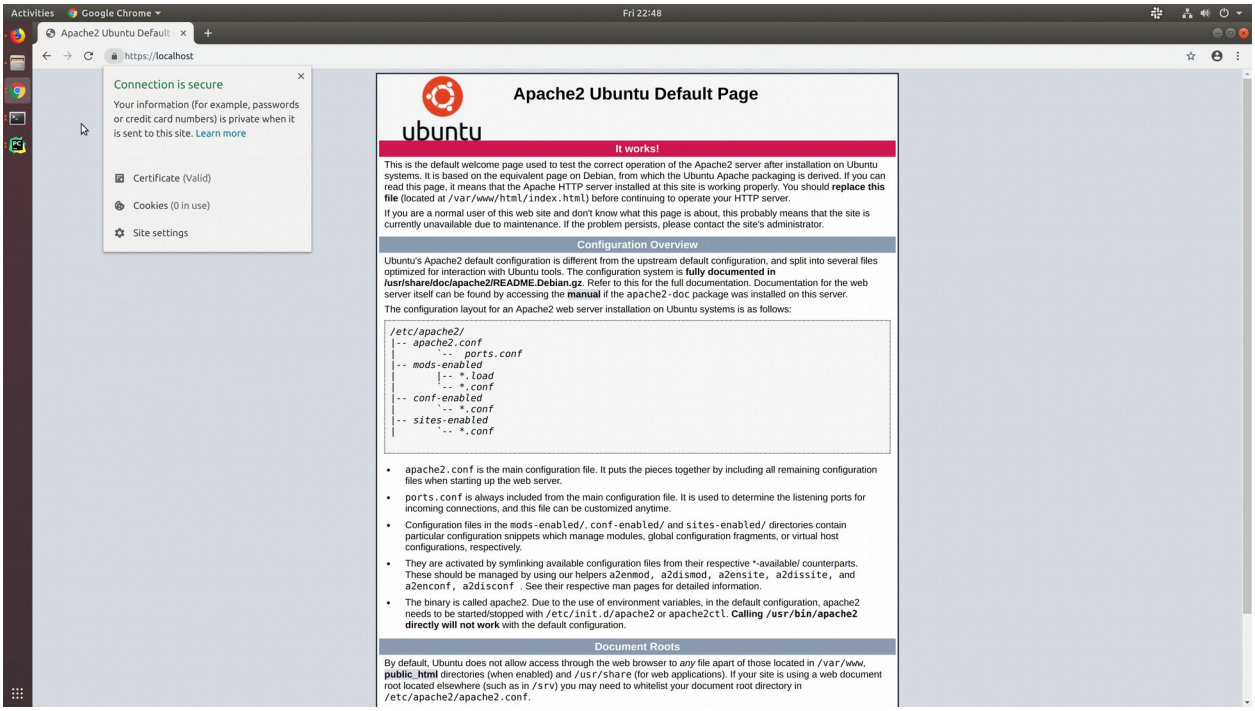
Although the server is providing secure data. The browser can not recognize the connection as secure. If we want to go to our address at this stage, we will see this



the reason is our certificate provider is not known to browser. Now we need to add out certificate to the trusted CA list. I have automated all the instruction for adding the certificate in CA list in “add_certificate_CA_list.sh”

This file need root permission. Running this file will add our certificate to the system CA list and also to the databases of Chrome and Mozilla Firefox.

Now if we access https://localhost we will see the browser will show the connection as secure.



https://thomas-leister.de/en/how-to-import-ca-root-certificate/

This article gives very nice instruction on how to add the self assigned certificate in Trusted CA list.

Future Work

My primary plan was to write a program which will run without asking the user for permission. But unfortunately I couldn’t figure out any way to get root privilege from the script.

References

- https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04
- https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-18-04
- https://thomas-leister.de/en/how-to-import-ca-root-certificate/
- https://www.rosehosting.com/blog/how-to-generate-a-self-signed-ssl-certificate-on-linux/
- https://www.digicert.com/ssl/
- https://www.websecurity.symantec.com/security-topics/how-does-ssl-handshake-work