Assignment 3. Securing server communications with HTTPS

HTTPS provides a secure communications channel between the server and a client. Description of HTTPS can be found in many popular textbooks, such as "Network Security Essentials" by William Stallings. The following link presents an overview of HTTPS: https://www.tutorialsteacher.com/https

In this assignment, the objective is to emulate a raspberry pi device running a client (Postman). We will install Raspbian OS on a VM and enable HTTPS on the REST API server used in Assignment 1. For this purpose, we will use a self-signed certificate, an easy and inexpensive method to enable HTTPS in test environments.

Part 1. Raspberry pi emulation

1. Raspbian Installation.

Raspbian is a Debian-based computer operating system for Raspberry Pi. It is a free operating system based on Debian, optimized for the Raspberry Pi hardware. Raspbian comes with over 35,000 packages: precompiled software bundled in a friendly format for easy installation on your Raspberry Pi. It has Python, Scratch, Sonic Pi, Java and more. Raspbian can be installed using the following URL: https://www.raspberrypi.org/downloads/raspberry-pi-desktop/

Information on how to install it on a VM can be found at https://thepi.io/how-to-run-raspberry-pi-desktop-on-windows-or-macos/

[*Make sure to select 'Install' option while booting up for the first time as running it as a live image causes storage issues.*]

[D1] Take a screen capture of the VM running Raspbian OS

2. Snapd Installation on Raspbian.

Snap is a software deployment and package management system developed by Canonical for the Linux operating system. The packages, called snaps, and the tool for using them, snapd, work across a range of Linux distributions allowing distribution-agnostic upstream software packaging. To install snapd, use the following commands:

```
$ sudo apt update
$ sudo apt install snapd
$ sudo reboot
```

More information on downloading Snapd can be found at: https://snapcraft.io/docs/installing-snap-on-raspbian

3. Postman Installation on Raspbian.

We would be using Postman as we had done in the previous assignments to signup to the server and to change the state of the server.

To install Postman on Raspbian use the following command:

```
$ sudo snap install postman
```

To start Postman, type the following command and **[D2] Take a screen capture of Postman running on Raspbian.**

```
$ postman
```

Information about this can be found on:

https://linux4one.com/how-to-install-postman-on-debian-9/

https://snapcraft.io/install/postman/raspbian

Part 2. Enabling HTTPS

1. Self-Signed Certificate creation for HTTPS:

To enable HTTPS connection between the client and the server we would need to create a self-signed certificate using **OpenSSL**, see https://flaviocopes.com/express-https-self-signed-certificate/

Run the following commands on Linux or Mac:

```
$ openssl req -nodes -new -x509 -keyout server.key -out server.cert
```

Type the following command and verify the contents of the certificate. **Take a screen capture of the output [D3]**.

```
$ openssl x509 -text -noout -in server.cert
```

2. Modify the server code (host computer):

The code of the server used in Assignment 1 should be modified for supporting HTTPS, information regarding how to use HTTPS in Express is available in the following links:

https://flaviocopes.com/express-https-self-signed-certificate/
https://timonweb.com/posts/running-expressjs-server-over-https/
[D4] Report the modifications to the server code

[*Note: The same code from the provided links would not work and would need to be slightly modified*]

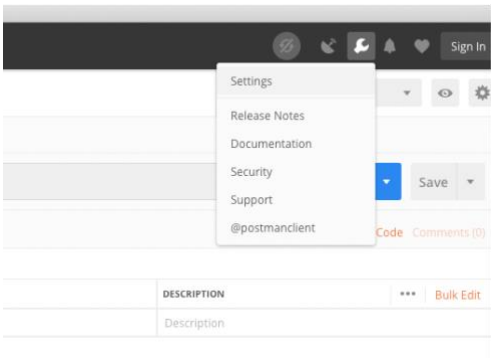
To build the server on the host computer, type:

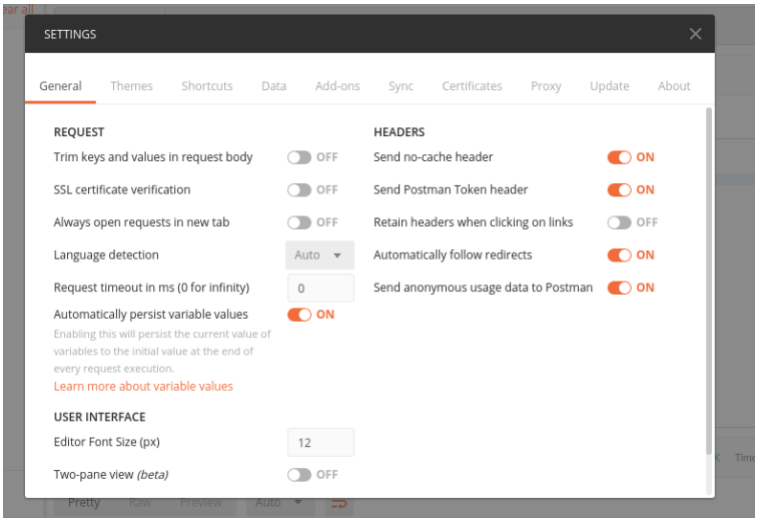
$ npm install
$ node app.js

Part 3. Endpoints test using Postman (Raspbian)

We will test the endpoints with the Postman instance running on Raspbian. The server is running on the host computer, so you will use the host's IP address for POST and GET requests.

1. Since the server is using a self-signed certificate, the "SSL certificate verification" option must not be selected, as shown below.





2. Test the following endpoints and verify the responses using Postman

https://<host's IP address>:3000/save_state **[D5.A] Send the state "ON" to the server and take a screen capture of the output**.
https://<host's IP address>:3000/last **[D5.B] Take a screen capture of the output.**

Report:

A document with deliverables D1-D5. 20 points each

D1: Raspbian successfully installed on a VM.
D2: Postman successfully installed and running on the Raspbian VM.
D3: Self-signed certificate generation.
D4: Server code used to enable HTTPS on a server.
D5: Various tests performed using HTTPS communication via Postman.