**Securing Node-Red with SSL and Username Authentication**

https://192.168.1.36:1880

Out of the box your connection to node-red is insecure.

If you are exposing node-red to the Internet or just want a more secure installation then you can:

* Enforce an SSL connection.
* Require Username and Password Authentication

**Note:**

When configuring multiple security settings I try to configure one security setting at a time and test it before configuring the next.

**Configuring SSL**

To do that you will need to install a certificate and key on your node-red server.

In this tutorial we will be using a self signed certificate which we will create ourselves using **openssl** and I will be using**raspberry pi** to host node red.

**Creating a Self Signed Certificate**

1.Create a private key

openssl genrsa -out node-key.pem 2048

2. Create a certificate Request

openssl req -new -sha256 -key node-key.pem -out node-csr.pem

You will need to fill out a form the most important entry is near the end and is the**common name** field.

This should be the **FQDN** of the server hosting nod-red or the IP address. I used **raspberrypi.home**

3. Sign the Certificate with the Private key to create a self signed Certificate

openssl x509 -req -in node-csr.pem -signkey node-key.pem -out node-cert.pem

**Note:** I created my files in sub folder of my .node-red folder called nodecerts.

**Editing the Node-Red Settings File**

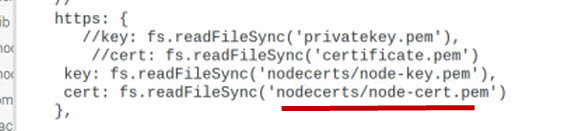
**/usr/lib/node-modules/node-red/**

un-comment the line

var fs=require("fs")

Which is near the top of the file.

In the **AdminAuth** section un-comment the https section and edit as shown in the screenshot below:



There is another setting

**requireHttps: true,**

Note: there should be a comma at the end

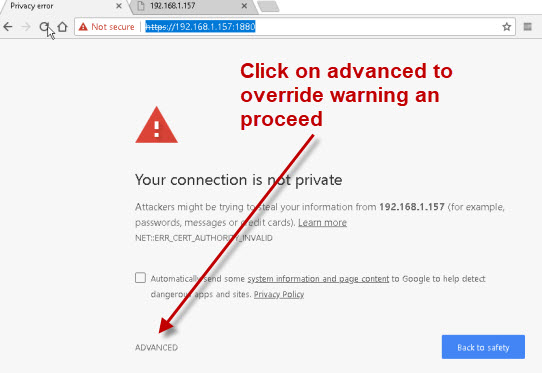
This should cause a redirect to **https** if you try connecting to **http**.

**Note**:I didn’t get this to work with SSL so you may need to skip this.

You can now start node-red using:

node-red -s mysettings.js

Now when you connect using the browser you should get a certificate warning which you can override.



**Username and Password Authentication**

The Node-Red Editor and Admin API support two types of authentication:

* username/password credential based authentication
* *OAuth/OpenID authentication -since Node-RED 0.17:*

On node-red there are three places were you can configure/require authentication.

* The Admin console- Node red editor
* Nodes
* Static Pages

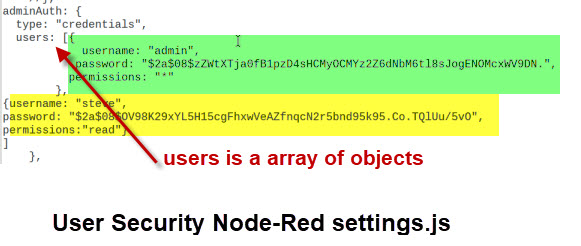
In this section we will look at username and password authentication for the admin console.

The default settings file has a section called **Securing node red** which is commented out and serves as a template.



The default entry shows the userAdmin with a password of password (hashed) and all permissions (\* wildcard).

You can create additional users by copying the users section and editing accordingly.



**Note:** The password hash for steve is different than that of admin,but the actual password is the same. This is how the **bcrypt** algorithm works

Steve has got read permissions which means that he cannot edit or create new flows.

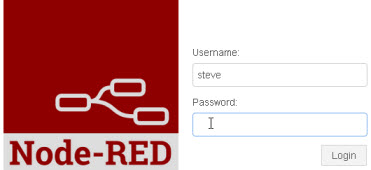
**Generating Passwords**

You can create passwords for use in the settings file by using the admin command line **hash-pw** command as follows:

node-red-rcreate-password

If you right click on the password you can copy it using CRT+C.

When you try to connect to the admin page you should get a login screen.



You can also use online**bcrypt** tools for creating the passwords like this [one](https://www.dailycred.com/article/bcrypt-calculator).

**Securing The Dashboard**

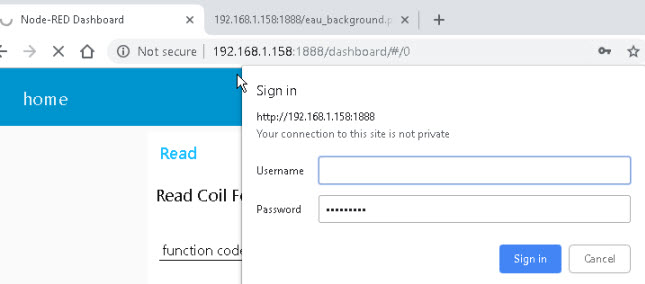
To secure the node-red dashboard use the **httpNodeAuth** setting.

Only a single user account is allowed.

httpNodeAuth: {user:"fred",pass:"$2a$04$3gkGX/Q4VZ//F37kWvSU9eE9EM1WO2rdWk1oj/kfXIbeBON5eA56S"},

You create the passwords using the **hash-pw** command line tools as before.

When you try and access the page you should be prompted to login.



**Securing Static Pages**

Node-red can serve static web pages. These web pages are served from the **usr/lib/node-modules/public** folder by default.

You can change to location by using the **httpStatic** setting in the settings file.

Generally you would want to move it to a folder in your home directory e.g.

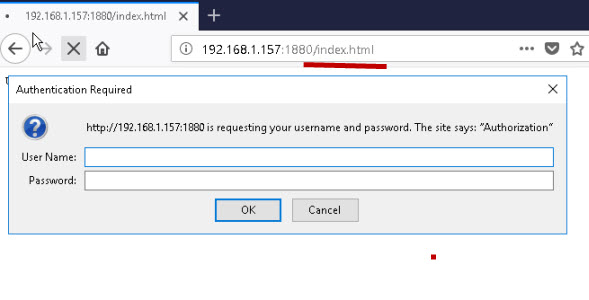
**httpStatic** **‘/home/steve/.node-red/node-red-static/’**

You can secure this folder using the **httpStaticAuth** setting.

httpStaticAuth:{user:"fred",pass:"$2a$04$3gkGX/Q4VZ//F37kWvSU9eE9EM1WO2rdWk1oj/kfXIbeBON5eA56S"},

You create the passwords using the **hash-pw** command line tools as before.

When you try to access the page you should be prompted to login.

  
**Note:** Only a single account is possible. It also appears to clash with the **httpNodeAuth** setting which is used to secure the dashboard.

If using both use the same username and password for both.