CoovaChilli Hotspot Captive Portal using FreeRADIUS and MySQL on Raspberry Pi

Cervin Cartilla

May 17, 2017

Installation of **CoovaChilli** powered **Hotspot Captive Portal** using **MySQL and FreeRADIUS** on **Raspberry Pi**

Notes:

- Raspberry Pi
- 16GB+ SD card with fresh install Raspbian OS
- Router for Access Point
- Internet connection

Make sure we all the latest updates installed in Raspbian. Run:

```
sudo apt-get update
```

To make sure Coova will run without any problems, we will install the dependencies first. To do so, run the following commands:

```
sudo apt-get install -y -f haserl gengetopt devscripts bash-completion libtool libjson-c-dev libssl-dev autoconf automake
```

To apply new settings restart the Raspbian:

```
reboot
```

I. Downloading and Installing the CoovaChilli Project

1. To clone the projects to your directory, use the command:

```
git clone https://github.com/coova/coova-chilli.git
```

2. Once you cloned you can find a copy of CoovaChilli project in the working directory. Confirm if there's a folder named **coova-chilli**. Then move the folder in the /usr/src directory, use the command:

```
mv /home/pi/coova-chilli /usr/src
```

3. Then go inside of the folder coova-chilli by changing directory, run the command:

```
cd /usr/src/coova-chilli
```

4. Then run the command:

```
sudo dpkg-buildpackage –us -uc
```

Once you run the command it will take some time until it's finalized. After it's done run the generated .deb file inside the folder /usr/src by running this command:

```
sudo dpkg –i coova-chilli_1.3.0_armhf.deb
```

The command above took around a few minutes to complete. Wait for the installation to finished.

II. Downloading and Installing the MySQL and FreeRADIUS

1. Install the packages that you will be using in conjunction with CoovaChilli and for the database of the user.

```
sudo apt-get install mysql-server freeradius freeradius-mysql
```

III. Configuration of Network Interface

1. This section is to set up your network interfaces appropriately so that one interface connects to the internet and the other interface is set up so that it can be used by CoovaChilli to manage your hotspot clients, run the command:

```
sudo nano /etc/network/interfaces
```

Modify the file to look like this: (line that starts with # is a comment line)

```
auto lo
iface lo inet loopback

#eth0 is the wired interface that will be managed by CoovaChilli
#The ip address listed below will be entered into the file
#/etc/chilli/defaults and assigned to the variable
#HS_UAMLISTEN=10.1.1.1 and the network address will be #assigned
to HS_NETWORK=10.1.1.0, HS_LANIF=eth0

auto eth0
iface eth0 inet static
        address 10.1.1.1
        netmask 255.255.255.0
        network 10.1.1.0

#The primary internet connection
#Given that eth1 is the primary internet connection then #later
in the file /etc/chilli/defaults we set the variable
#HS_WANIF=eth1

auto eth1
```

When you're done, restart the networking service:

```
sudo /etc/init.d/networking restart
```

IV. Configure MySQL

1. Switch to root user first.

```
sudo su
```

2. Create a MySQL database called 'radius':

```
#mysql –u root –p
mysql> create database radius;
mysql> quit
```

3. Create the tables in the new radius database you just set up, run the commands:

```
#mysql –u root –p radius </etc/freeradius/sql/mysql/schema.sql
#mysql –u root –p radius </etc/freeradius/sql/mysql/admin.sql
#mysql –u root –p radius </etc/freeradius/sql/mysql/nas.sql
```

4. Create a user. This is a test user who will login to your CoovaChilli hotspot account later.

```
#echo "insert into radcheck (username, attribute, op, value)
values ('user1','Cleartext-Password',':=','pass1');" | mysql
–u root –p radius
```

V. Configure FreeRADIUS

The first step is to configure FreeRADIUS to use the MySQL database you just set up.

1. Edit the main radius configuration file:

```
nano /etc/freeradius/radiusd.conf
```

Uncomment the line:

```
$INCLUDE sql.conf
```

2. Edit also this file to make MySQL work:

```
nano /etc/freeradius/sites-available/default
```

This file is organized into sections of code and there are three sections you need to modify. Find the following blocks of code and uncomment the sql line in each.

```
authorize{
sql
}

accounting{
sql
}

session{
sql
}
```

3. Now that you have FreeRADIUS configured to work with MySQL, to ensure it is configured properly. First stop FreeRADIUS daemon.

```
#/etc/init.d/freeradius stop
```

Then restart FreeRADIUS in debug mode:

```
#freeradius -X
```

At this point you'll see reams of output filling up your screen. What's important here is to look at the end and check for a line that says **"Ready to process requests"**. If you see that then there's no error. If not then check the reams of output as it will tell you where the error occurred. Check to make sure you didn't forget to uncomment something or you didn't make a typo.

Now that FreeRADIUS is processing requests you need to make sure it will authenticate the test user you entered previously, that is, **user1** with its associated '**pass1**'. Open up another terminal and enter the following:

```
#radtest user1 pass1 localhost 0 testing123
```

If all goes well, you should see output that says: **"rad_recv: Access-Accept packet from host 127.0.0.1…."**. If not, check to make sure you added the user and password correctly and that you've followed the correct sequence above to allow FreeRADIUS to read the MySQL database.

Now you can close the extra terminal you've opened and in the first terminal hit <ctrl-C> to stop freeradius in debug mode. Now restart the freeradius daemon:

```
#/etc/init.d/freeradius start
```

VI. Configure CoovaChilli

1. Go back to into the coova-chilli source code directory:

```
#cd /usr/src/coova-chilli
```

2. Configure the source file with following options.

```
/usr/src/coova-chilli# ./configure --prefix=/usr
--mandir=\$${prefix}/share/man --infodir=\$${prefix}/share/info \
--sysconfdir=/etc --localstatedir=/var --enable-largelimits \
--enable-binstatusfile --enable-statusfile --enable-chilliproxy \
--enable-chilliradsec --enable-chilliredir --with-openssl --with-curl \
--with-poll --enable-dhcpopt --enable-sessgarden --enable-dnslog \
--enable-ipwhitelist --enable-redirdnsreq --enable-miniconfig \
--enable-libjson --enable-layer3 --enable-proxyvsa --enable-miniportal \
--enable-chilliscript --enable-eapol --enable-uamdomainfile \
--enable-modules --enable-multiroute
```

```
root@raspberrypi:/usr/src# cd coova-chilli-1.3.0
root@raspberrypi:/usr/src/coova-chilli-1.3.0# ./configure  --prefix=/usr --mandi
r=$${prefix}/share/man \
> --infodir = \ $${prefix}/share/info \
> --sysconfdir=/etc --localstatedir=/var --enable-largelimits \
> --enable-binstatusfile --enable-statusfile --enable-chilliproxy \
> --enable-chilliradsec --enable-chilliredir --with-openssl --with-curl \
> --with-poll --enable-dhcpopt --enable-sessgarden --enable-dnslog \
> --enable-ipwhitelist --enable-redirdnsreq --enable-miniconfig \
> --enable-libjson --enable-layer3 --enable-proxyvsa --enable-miniportal \
> --enable-chilliscript --enable-eapol --enable-uamdomainfile \
> --enable-modules --enable-multiroute
configure: WARNING: you should use --build, --host, --target
configure: WARNING: invalid host type:  13055{prefix}/share/info
configure: error: expected an absolute directory name for --infodir: =
```

There are errors were expected so you need to set the compatibility level for debhelper otherwise you'll get something like:
…: No compatibility level specified in debian/compat
… :The package will soon FTBFS; time to fix it!
… :Compatibility levels before 5 are deprecated (level 1 in use)

3. Failing to set the compatibility level may result in a .deb file that's incomplete, unstable or unusable. Therefore let's set the compatibility level to 9:

```
/usr/src/coova-chilli$ echo 9 > debian/compat
```

4. Now we need to modify the /usr/src/coova-chilli/debian/rules

```
$sudo nano /usr/src/coova-chilli/debian/rules
```

On line 54 of the file under install:build, you need to replace the following line:

```
$(MAKE) DESTDIR=$(CURDIR)/debian/tmp install

With:

$(MAKE) DESTDIR=/install
```

This ensures the necessary files are put into /etc/chilli/directory and not in the directory specified in the original line. Failing to do this will cause an error.

5. Go to /usr/src/ directory:

```
cd /usr/src
```

6. Now that you're compiling your code from source. You will download again another source code.

```
/usr/src $ sudo wget
http://downloads.sourceforge.net/project/haserl/haserl-
devel/haserl-0.9.30.tar.gz
```

Haserl is a light-weight program to create cgi web scripts.

7. Enter the haserl directory

```
/usr/src $ cd haserl-0.9.30
```

8. Configure the file:

```
/usr/src/haserl-0.9.30 $ ./configure
```

9. Now we make and install the file:

```
/usr/src/haserl-0.9.30 $ sudo make && sudo make install
```

Haserl should now be installed.

10. Enable the service of CoovaChilli so it will start. It is turned off by default. To enable it, simply edit the following file:

```
# nano /etc/default/chilli
```

Change the first line from 0 to 1:

```
START_CHILLI = 1
```

11. Next edit the main chilli configuration file:

```
# nano /etc/chilli/defaults
```

This is the file referred to previously when you set up your network interface. The items you're changing should be the same in the /etc/network/interfaces file.

```
HS_WANIF = eth1                 # From our /etc/network/interfaces file
HW_LANIF = eth0                 # From our /etc/network/interfaces file
HS_NETWORK = 10.1.1.0           # From our /etc/network/interfaces file
HS_UAMLISTEN = 10.1.1.1          # From our /etc/network/interfaces file
HS_DNS1 = 8.8.8.8                # Set it to google because I know it works
HS_UAMALLOW = 10.1.1.0/24
HS_RADSECRET = testing123        # This doesn't need to change but you might
want to change this later.
                                  # You'll just need to change it in
/etc/freeradius/clients.conf as well
HS_LOC_NAME = "HotSpot"          # Change this to what you'd like to appear
on your login page
```

VII. Configure Firewall

1. Our next step is to configure the firewall using iptables. CoovaChilli comes preconfigured to do the heavy lifting for you but we do need to add a couple of entries. Edit the following file:

```
$sudo nano /etc/chilli/up.sh
```

At the bottom of the file add the following line:

```
iptables –I POSTROUTING –t nat –o $HS_WANIF –j MASQUERADE
```

2. You installed earlier the haserl, now edit this file and add the path to haserl:

```
$sudo nano /etc/chilli/wwwsh
```

Go to the line where it says:

```
haserl = $(which haserl 2>/dev/null)
```

and replace it with:

```
haserl = /usr/local/bin/haserl
```

3. Restart all services to ensure all changes are made:

```
reboot
```

4. Once you've logged back into the Raspbian, you will need to start chilli.

```
$sudo /etc/init.d/chilli start
```

Okay. Your CoovaChilli Captive Portal should now be configured.

If you check ifconfig you should see a new **tun0** interface, then it is working.

```
$ifconfig
```

VIII. Setup the Router Access Point

To make CoovaChilli work, you need to turn you router into a bridge. In **DD-WRT** you need to do the following:

Disable the DHCP Server
Under Setup: Basic Setup
DHCP Server: Disable

Set your broadcast SSID
Under Wireless: Basic Settings
Wireless Network Name (SSID) Your-Hotspot
Wireless Channel: Auto

Disable the Security Mode
Under Wireless: Wireless Security
Security Mode: Disable

Disable the firewall
Under Security
Firewall Protection: Disable

Plug the Ethernet cable from your Pi into a LAN port on your router (not the WAN/internet port), this is the **eth0**. Then the other Ethernet cable from another source of internet connection, this is the **eth1.**

From another computer or Android/iOS devices wirelessly connect to the network you just set up. Start your web browser and it should bring up the Coova login page.

Remember you set up a test user called ‘user1’ with a password of ‘pass1’. Enter those into the user/password fields. If all is successful you should be able to access the internet through your hotspot. Now enter a non-existent user and password to see if it fails, it should.

IX. Setting Up Users and Setting Restrictions (Time and Traffic Limits)

To set the time and traffic limit of the user those are managed and setup through FreeRADIUS rather than CoovaChilli which means you'll be modifying our MySQL tables to set the restrictions.

1. Edit the following file:

```
$sudo nano /etc/freeradius/radius.conf
```

Uncomment the line so it reads:

```
$INCLUDE sql/mysql/counter.conf
```

2. Now you will edit the following file:

```
$sudo nano /etc/freeradius/sites-available/defaults
```

Within the authorize section of the file, you're going to uncomment sql; however, depending on the restrictions you want to apply you must also add them to this file. You're going to add four counters: noresetcounter, dailycounter, monthlycounter, expiration. All but expiration are found in /etc/freeradius/sql/mysql/counter.conf since expiration is a module.

authorize {
sql ← uncomment
noresetcounter ←add
dailycounter ←add
monthlycounter ←add
expiration ←add
daily ← uncomment this one
}
accounting {
daily ←uncomment
}

3. Now that you've modified this file you need to add some entries into the dictionary:

```
$sudo nano /etc/freeradius/dictionary
```

Add the following lines:

ATTRIBUTE	Daily-Session-Time	3000	integer
ATTRIBUTE	Max-Daily-Session	3001	integer
ATTRIBUTE	Monthly-Session-Time	3002	integer
ATTRIBUTE	Max-Monthly-Session	3003	integer
ATTRIBUTE	Max-All-Session-Time	3004	integer
ATTRIBUTE	Max-All-Session	3005	integer

4. To deal with the expiration restriction you need to add some code:

```
$sudo nano /etc/freeradius/sql/mysql/counter.conf
```

```
sqlcounter expiration {
               count-attribute = "Acct-Session-Time"
               counter-name = "Max-Allowed-Session"
               check-name = "Expiration"
               sqlmod-inst = "sql"
               key = "User-Name"
               reset = "never"
               query = "SELECT
IFNULL(TIME_TO_SEC(TIMEDIFF(NOW(), \
               MIN(AcctStartTime))),0) FROM radacct WHERE \
                    UserName='${key}' ORDER BY AcctStartTime
LIMIT 1;"
}
```

5. Now that you've got your system configured you can now proceed to add new users along with restrictions.
If you want to add a new user called "**mhelvin**" with an associated password "**mhelvinpogi**" we do the following:

```
#mysql –u root –p

#echo “insert into radcheck (username, attribute, op, value)
values (‘mhelvin’, ‘Cleartext-Password’, ‘:=’,
‘mhelvinpogi’);” | mysql –u root –p radius
```

Now one of the benefits of using MySQL is that immediately after you added **mhelvin**, you can now login as **mhelvin** without restarting any of your services. You can add as many or as few of the restrictions provided below to any or all of your users. Add the desired restrictions to the appropriate users in the MySQL database.

Example 1:
This will set the maximum time that **kelvin** can access the internet to 10800 seconds or 3 hours:

```
#echo “insert into radcheck (username, attribute, op, value) values (‘kelvin’, ‘Max-Daily-Session’, ‘:=’, 10800);” | mysql –u root –p radius
```

Example 2:
The following will limit when **user1** can logon to the internet. Note that **user1** can only access the internet between 9.00am and 6.00pm in the example.

```
# echo “insert into radcheck (username, attribute, op, value) values (‘user1’, ‘Login-Time’, ‘:=’, ‘Al0900-1800’);” | mysql –u root –p radius
```

Example 3:
chris is only allowed to be logged on the system once. If you set this to 2 then **chris** can logon to the system twice simultaneously. This prevents people from sharing their user name and password with their friends and each of them to login at the same time.

```
# echo “insert into radcheck (username, attribute, op, value) values (‘chris’, ‘Simultaneous-Use’, ‘:=’, 1);” | mysql –u root –p radius
```

Example 4:
cervin’s access expires on August 26, 2017 at 8.00pm.

```
# echo “insert into radcheck (username, attribute, op, value) values (‘cervin’, ‘Expiration’, ‘:=’, ‘August 26 2017 20:00’);” | mysql –u root –p radius
```

Example 5:
gerald will be rejected no matter what.

```
# echo “insert into radcheck (username, attribute, op, value) values (‘gerald’, ‘Auth-Type’, ‘:=’, ‘Reject’);” | mysql –u root –p radius
```

Example 6:
ericson forgets to logout and therefore has an idle session and you want to automatically log **ericson** off if the session is idle longer than 120 seconds:

```
# echo “insert into radreply (username, attribute, op,
value) values (‘ericson’, ‘Idle-Timeout’, ‘=’, 120);” |
mysql –u root –p radius
```

Note: the equal sign is an ‘=’ sign and not a ‘:=’, the time is listed in seconds and the entry is into the table radreply and not radcheck.

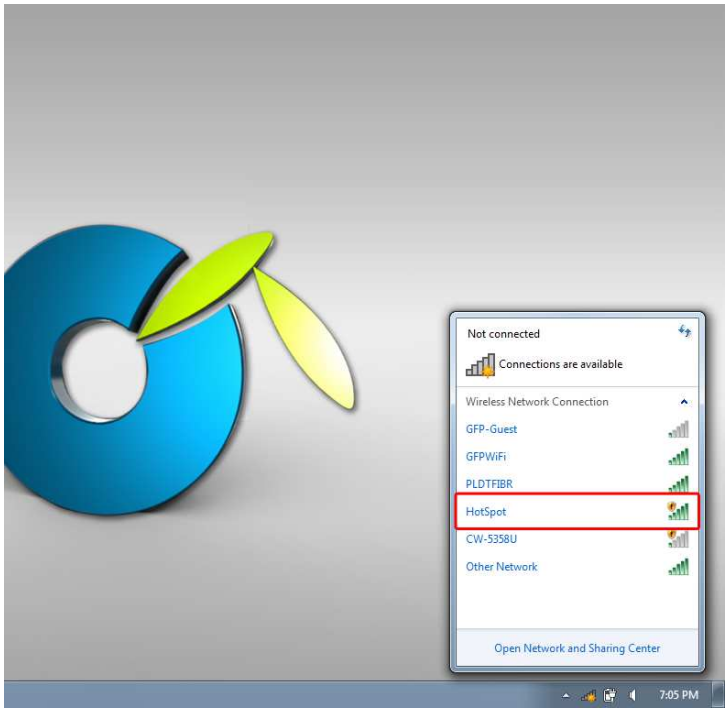
Example 7:
You want to limit **marco**’s sessions to 10 minutes each. In other words, after **marco** has been logged in he will be automatically logged out after 10 minutes. If **marco** logs out and back in, he will get another 10 minutes.

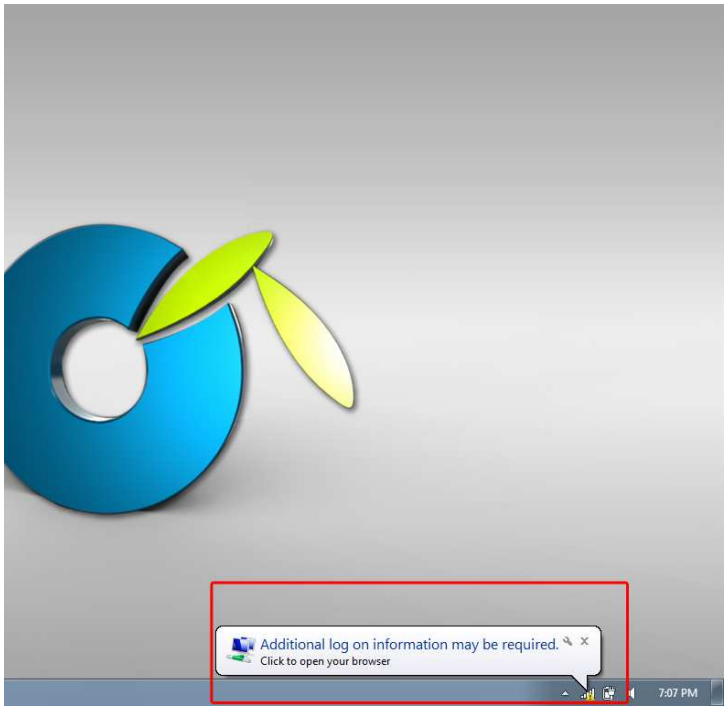
```
# echo “insert into radreply (username, attribute, op,
value) values (‘marco’, ‘Session-Timeout’, ‘=’, 600);” |
mysql –u root –p radius
```

X. Connecting to the CoovaChilli Hotspot

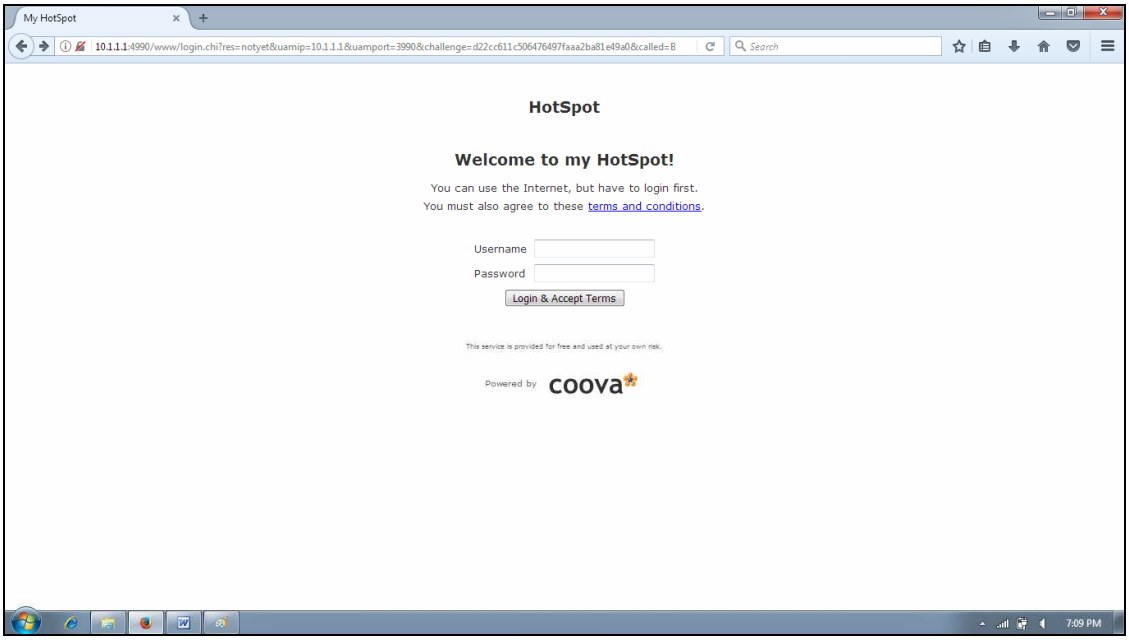
1. Using Windows

Connect to the wireless network.

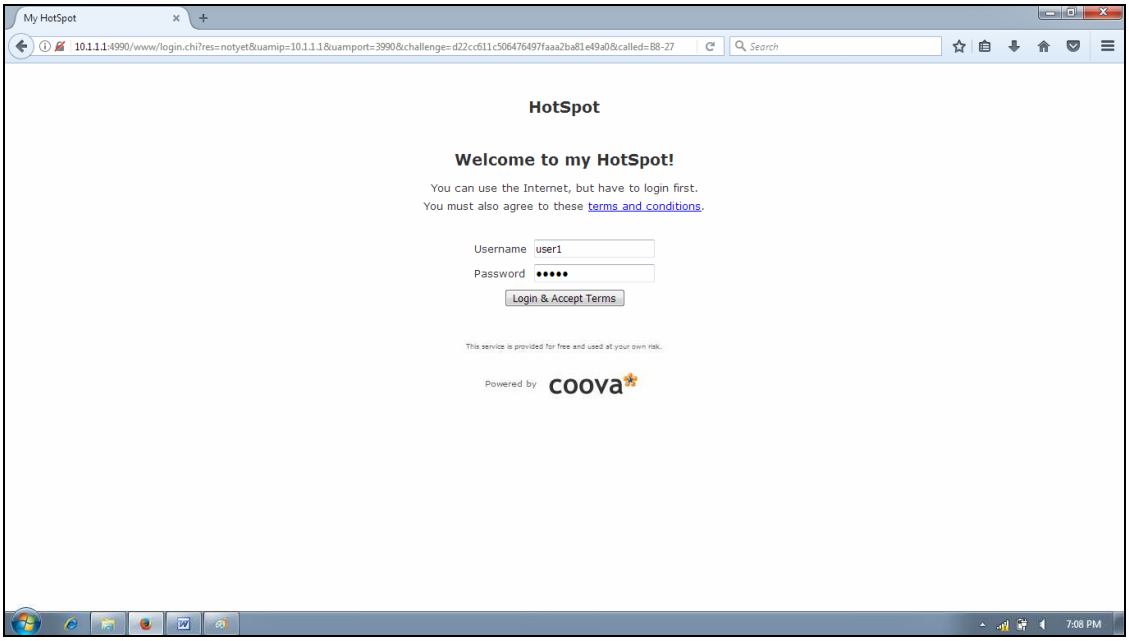




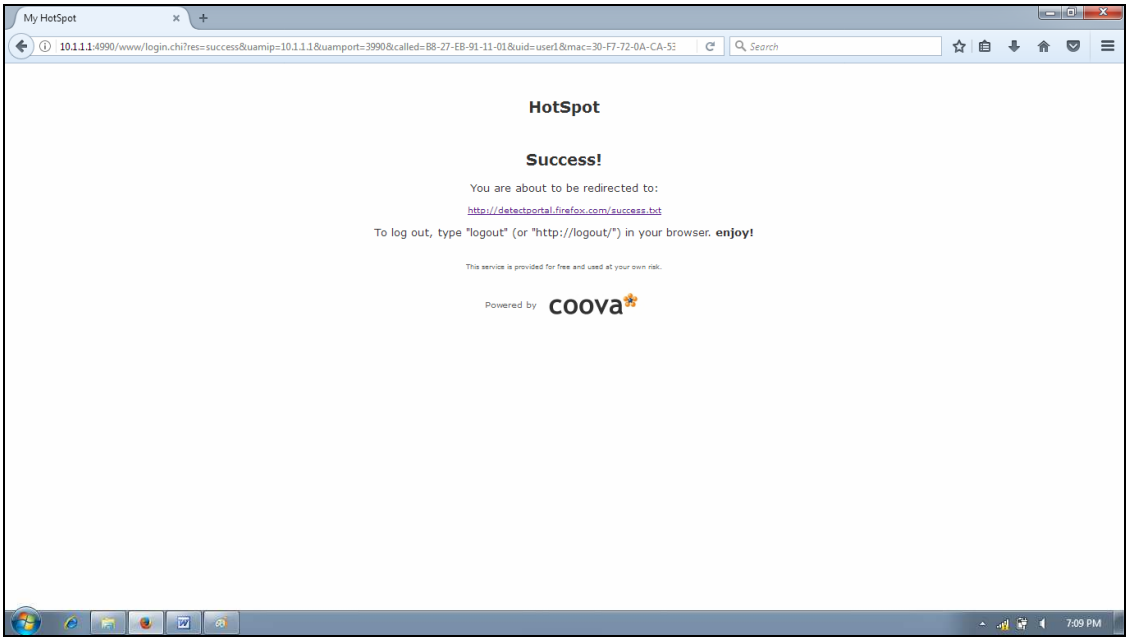
Open a browser search any website and you will automatic redirect to CoovaChilli Hotspot Login Page.



Login the user that you created earlier the ‘user1’ ‘pass1’

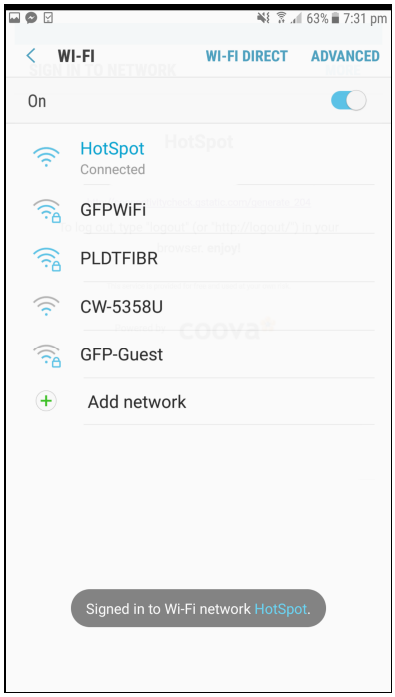


After a successful login, you can now have an internet connection.

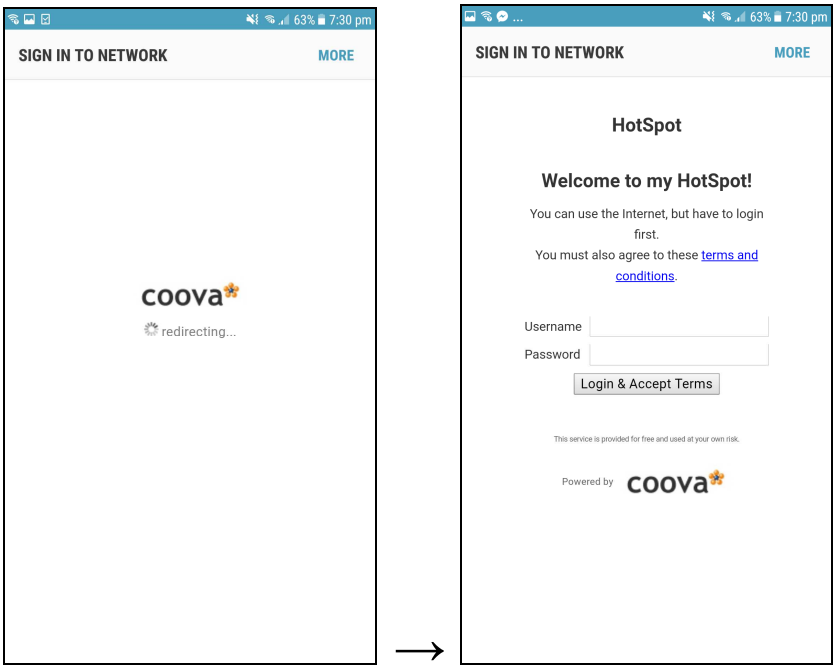


2. Using Android devices

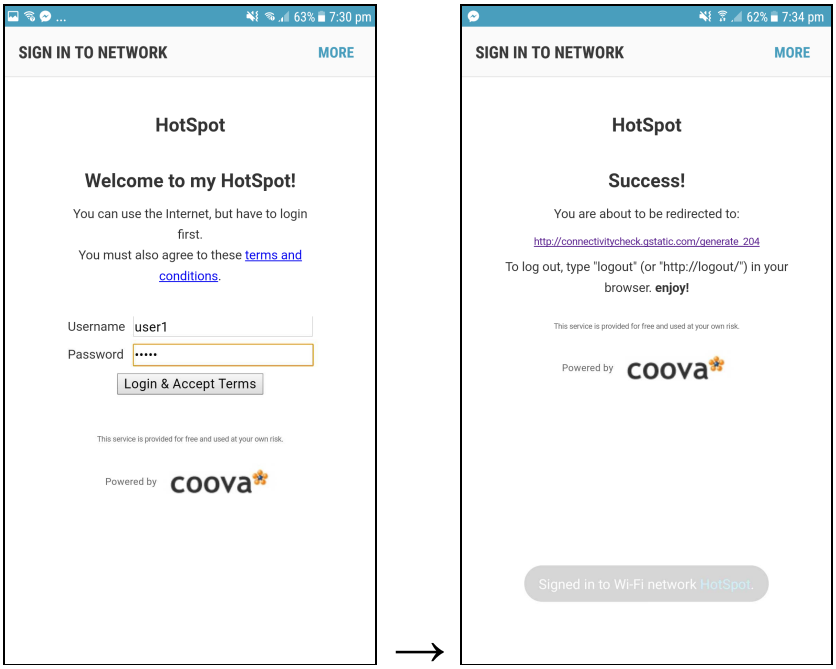
Connect to the wireless network



Once you’re connected you will automatic redirect to CoovaChilli Hotspot Login Page.



Login the user that you created earlier the ‘user1’ ‘pass1’
After a successful login, you can now have an internet connection.



XI. References:

https://www.academia.edu/11287983/HOWTO_Install_CoovaChilli_on_a_Raspberry_Pi_using_Raspbian_FreeRADIUS_and_MySQL

https://citricks.net/install-coova-captive-portal/

http://raspberry-pi-thai.blogspot.com/2015/08/raspberry-pi-wifi-hotspot-coovachilli.html