Setting up a Raspberry Pi as a Headless Webserver

The following guide works for me. There may well be other ways to achieve similar results, and they may be better, more secure, more efficient than mine.

If you know a little bit about setting up your Pi in general then skip this page, but if you're not sure then here are a few things to do before we get started.

1. Download the latest version of the [Raspberry Pi software](#) (NOOBS 1.2 at the time of writing). I'm sticking with the default installation of Raspbian Wheezy.

2. Reboot the Pi and in the raspi-config window choose **Overclocking - Medium**. This gives you a boost in power but shouldn't cause any damage or instability.

3. Login as pi / raspberry.

4. If security is an issue for you then change your password (type **passwd**). If you're just playing around in the classroom then this might not be so critical, but bear in mind that the default username and password are widely known and will give anyone full admin access to your webserver.

   If your webserver is going to be on the Internet (and not just in school) then you MUST change the password. Better still, create a new user and disable the pi user. If you're not sure how to do this then google

5. Find out if you are going to be behind a proxy server. If you're in school, that's probably a "yes". If you're at home then it's probably "no". If you are going to be behind a proxy server then find out the ip address and port number of your proxy server.
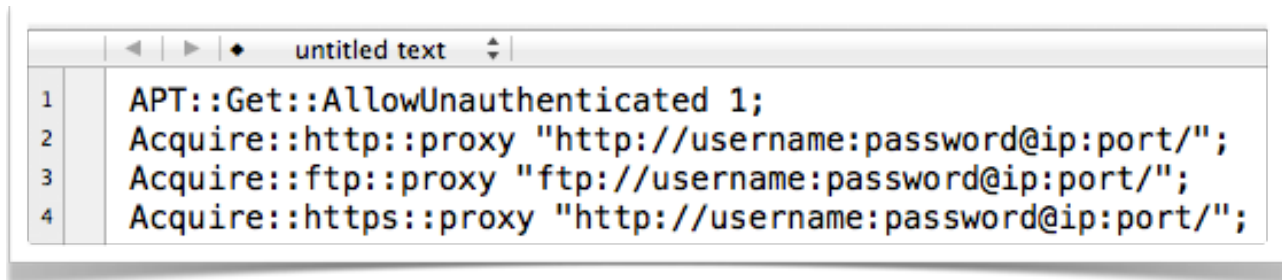
   Your normal school login will let you get to the Internet, through the proxy server. For security, it is best to get a new account set up just for the pi. It's possible that a student may dig deep enough to find your proxy username and password.

Set the apt-get proxy

```
sudo nano /etc/apt/apt.conf
```

Create a new proxy configuration file so you can download updates and packages. You can skip this task if you're not behind a proxy server.

Type the following 4 lines, substituting your username, password and the proxy server's address and port number.


```
1  APT::Get::AllowUnauthenticated 1;
2  Acquire::http::proxy "http://username:password@ip:port/";
3  Acquire::ftp::proxy "ftp://username:password@ip:port/";
4  Acquire::https::proxy "http://username:password@ip:port/";
```

Press CTRL-x to exit and save the file.

Check the proxy and update the system

```
sudo apt-get update
```

This will update your system software. If it doesn't work, you know you have problems connecting to the Internet. First stop would be to check for typos in the file from the step above.

Install apache2
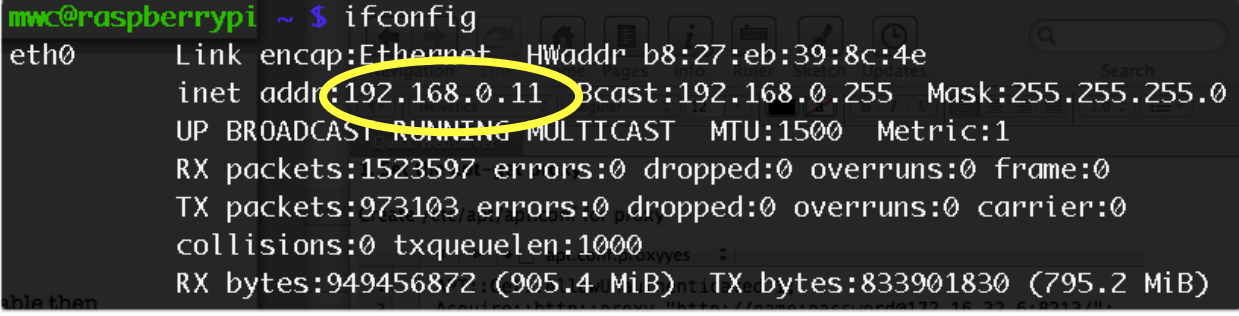
```
sudo apt-get install apache2
```

This will install both packages. You will be prompted to confirm at one point, just type 'y' and press enter.
Apache2 is the webserver software that will let you host webpages.

Check your webserver

```
ifconfig
```

Find out what your IP address is. If you're connected via an ethernet cable then look for the 'inet' address on en0

Go to a computer with a web browser, type in this IP address and you should see a page showing that your webserver is running.

Add users

```
sudo groupadd www

sudo useradd -g www -m -b /var/www student01
sudo passwd student01
```

The first line will create a new usergroup called "www".

The next line will create a new user called "student01".

-g www	adds the user to the www group.
-m	forces the creation of a home directory for this user.
-b /var/www	sets the home directory for that user to /var/www/<username>.

The third line will let you set the password for this user.

Repeat the second and third lines for each new user. If you are creating a small number of users (up to 15) you can do this manually. To create a bulk list of users isa bit more complicated.

Add users

```
su -- student01

cd ~
pwd
nano index.html
```

The first line will let you log in as student01 and test that the account works.

The second line will take you to student01's home directory (/var/www/student01).

The third line will tell you where you are (so you can check that it is correct).

The final line will create a blank file called index.html and let you edit it with the nano text editor. Create a simple web page, save it (CTRL-x) and test it by going to something like this:

192.168.0.11/student01

Type `exit` to drop back into your normal account.

And you're done! Your students can now use SSH (via putty, a free download for Windows) to log into your Raspberry Pi and can then use nano to create and edit web pages using HTML.

This isn't necessarily the easiest way to manage your webserver, but it's the minimum you need to get it up and running.

Install vsftpd

```
sudo apt-get install vsftpd
```

This will install vsftpd (Very simple FTP Daemon). This will allow users to transfer files to the server using FTP. This could be with an FTP client (there are many available) or it could be done straight from Dreamweaver or another web design package.

Configure vsftpd

```
sudo nano /etc/vsftpd.conf
```

Edit the standard vsftpd configuration file.

First, change the line "anonymous_enable=YES" to "anonymous_enable=NO". This will prevent users from logging in anonymously (without having to log in).

Next, remove the comment (#) at the start of the lines that say "local_enable=YES" and "write_enable=YES". This will allow users to log in and to make changes (upload files to the server).

You (and any users with an account on the Pi) can now connect using an FTP client, using the Pi's IP address and their login details. By transferring an existing web page or website to the Pi (/var/www/student01), the students can put their work 'online' within your local network, or even the world wide web.

If you use Adobe Dreamweaver and you use the "manage sites" function, you can tell Dreamweaver to connect to an FTP server (on your Pi) and upload the website with a simple click of a button. Serif Webplus and Kompozer (free, open-source) also feature FTP tools.