Managing a Web Server: Lab 8 - .htaccess & Additional Sites

michaelferrie@edinburghcollege.ac.uk

Introduction

In this lab you password protect a site with .htaccess, then host several websites on the server. This will require a Debian installation with Apache 2, no databases or php scripts are used.

Part 1: Create example.com

1.1 For this task we need to create a new website called example.com. Create the site directory as root, then change the owner to your non-root username (in this case my username is michael) then set permissions. If your non-root username was michael then you could use this exact command, but I'd guess it's not:

```
mkdir -p /var/www/html/example.com
chown -R michael:michael /var/www/html/example.com
chmod -R 755 /var/www/html
cd /var/www/html/example.com
```

1.2 Make a simple site for example.com, this will create a default index page for the site. vi index.html, add a screenshot of the completed html when you have created the index file. Careful for the closed tags, you can set the vim syntax highlighting to html by pressing ESC then entering `:set syntax=html`

```
<!DOCTYPE html>
<html>
        <body>
                <h1>Welcome to Example.Com</h1>
                        <p>All kinds of examples for sale</p>
                        <p>An exemplary place</p>
        </body>
</html>
```

Part 2: Create VirtualHost and Enable example.com

2.1 Make the VirtualHost File, copy one of the default files and give it a new name, in this case example.com.conf

```
cd /etc/apache2/sites-available
cp 000-default.conf example.com.conf
```

2.2 Now we have a nice file, let's dd out the comments, make example.com.conf look like this:

```
<VirtualHost *:80>
      DocumentRoot /var/www/html/example.com
      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

2.3 The final step is to enable the new site and reload Apache, then check that you can browse to the site created in step 1.2 (use a2dissite to turn off any other sites), the site should be available on http://127.0.0.1/example.com. Take a screenshot of the site running in the browser.

```
a2ensite example.com.conf
a2dissite 000-default.conf
systemctl restart apache2
```

Part 3: Create .htpasswd file and enable .htaccess

3.1 Create an htpasswd file and add a user, remember the password now, creating a password for the the user bob would be done like this:

```
htpasswd -c /etc/apache2/.htpasswd bob
```

3.2 Now create a secret area in the new website with a secret text file, verify you can navigate to these in the web browser:

```
mkdir -p /var/www/html/example.com/secret
touch /var/www/html/example.com/secret/secret_file.txt
chmod -R 755 /var/www/html
```

3.3 Create a .htaccess file which will tell Apache to use .htaccess and inside the .htaccess file tell Apache where to look for the password file:

```
vi /var/www/html/example.com/secret/.htaccess
```

```
AuthUserFile /etc/apache2/.htpasswd
AuthGroupFile /dev/null
AuthName "Please Enter Password - RESTRICTED AREA!"
AuthType Basic
Require valid-user
```

3.4 Edit the example.com.conf file in sites-available add the following:

```
<VirtualHost *:80>
      DocumentRoot /var/www/html
      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined
             <Directory /var/www/html/example.com/secret>
                   AuthType Basic
                   AuthName "Restricted Content"
```

```
                        AuthUserFile /etc/apache2/.htpasswd
                        Require valid-user
                </Directory>
</VirtualHost>
```

3.5 Finally enable mod-rewrite, disable the other sites and restart the server:

```
a2enmod rewrite
a2dissite 000-default.conf
a2dissite default-ssl.conf
systemctl restart apache2
```

3.6 Browse to the secret area of the site in a new private window, you should be prompted for a password now, log in as bob. If a username and password box appears you have successfully configured .htaccess

Part 4: HTML5 Sites / FileZilla

4.1 Download the three html sites tar file from the Moodle, there is a fashion photography site, a recruitment company website and a marketing website. Create three new VirtualHost entries for these websites and host them as marketing.com, fashion.com and recruitment.com.

4.2 Edit the VirtualHost file for each of the sites and host them on ports 8001, 8002 and 8003.

CHALLENGE: Password protect the site on 8001 with .htaccess, and create a self signed certificate for the site on port 8002 - hint, you'll need to adapt the virtual host file to point to the certificate. This challenge task will be very helpful when you attempt the coursework.