

H16S35 - Managing a web server

Topic 8

- .htaccess
- hosting additional sites

michael.ferrie@edinburghcollege.ac.uk



CONTENTS

01 **.htaccess**

What is .htaccess
Common .htaccess uses

02 **.htpasswd**

Creating .htpasswd files
.htpasswd examples

03 **.htaccess Apache Configuration**

Prepare apache for .htaccess

04 **Additional Sites**

Creating Additional Sites
Example Site Directory

05 **Q and A**

Any Questions

.htaccess

michael.ferrie@edinburghcollege.ac.uk

What is .htaccess?

- .htaccess (hypertext access) file is a configuration file supported by web servers, used for website-access issues, like URL redirection, URL shortening and access control
- The 'dot' before the file name hides the file in Unix-based environments
- A site could have more than one .htaccess file, the files are placed inside the www root directory
- .htaccess files act as a subset of the server's global configuration file for the directory that they are in

Why `.htaccess`

- '`.htaccess`' is the full filename, it is not a file extension
- Do not create a file called, '`file.htaccess`', only
- '`.htaccess`'.
- This file will take effect when placed in any directory which is loaded via the server
- The file will take effect over the directory it is placed in and all files and subdirectories within the specified directory

.htaccess

.htaccess example contents

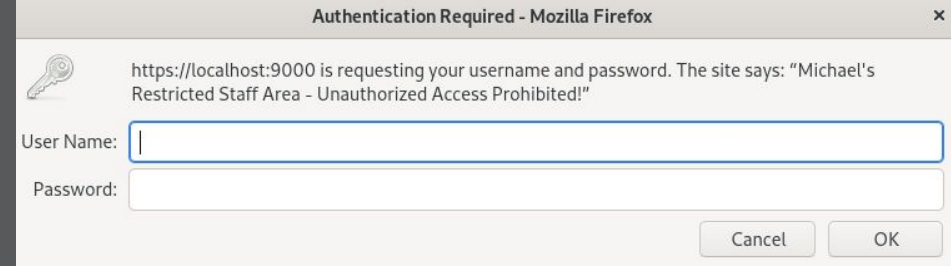
```
AuthUserFile  
/usr/local/bob/safedirectory/.htpasswd  
AuthGroupFile /dev/null  
AuthName "Please Enter Password"  
AuthType Basic  
Require valid-user
```


.htpasswd

michael.ferrie@edinburghcollege.ac.uk

Password Protection

- Password protection and authentication systems offered Apache are probably the most important use of .htaccess files
- Very easily, we can password protect a directory (or multiple) of a web site which require a username and password to access
- The login procedure for these secure directories is handled automatically by the web browser using a pop-up login interface (you've probably seen these before)
- Passwords are also encrypted which ensures login credentials are kept secure



Creating an htpasswd file

- The location of the file can be anywhere on the server, use the following command to create it, specifying the username after the command:

```
htpasswd -c /home/username/.htpasswd bob
```

- The contents of the `.htpasswd` file will look like this

```
bob:$apr1$FaPCZHMe$jYiw5.9UevKx25pBH4AsT
```

.htaccess apache configuration

michael.ferrie@edinburghcollege.ac.uk

Modify VirtualHost for .htpasswd

```
<VirtualHost *:80>

    DocumentRoot /var/www/html/example.com

    ErrorLog ${APACHE_LOG_DIR}/error.log

    CustomLog ${APACHE_LOG_DIR}/access.log combined

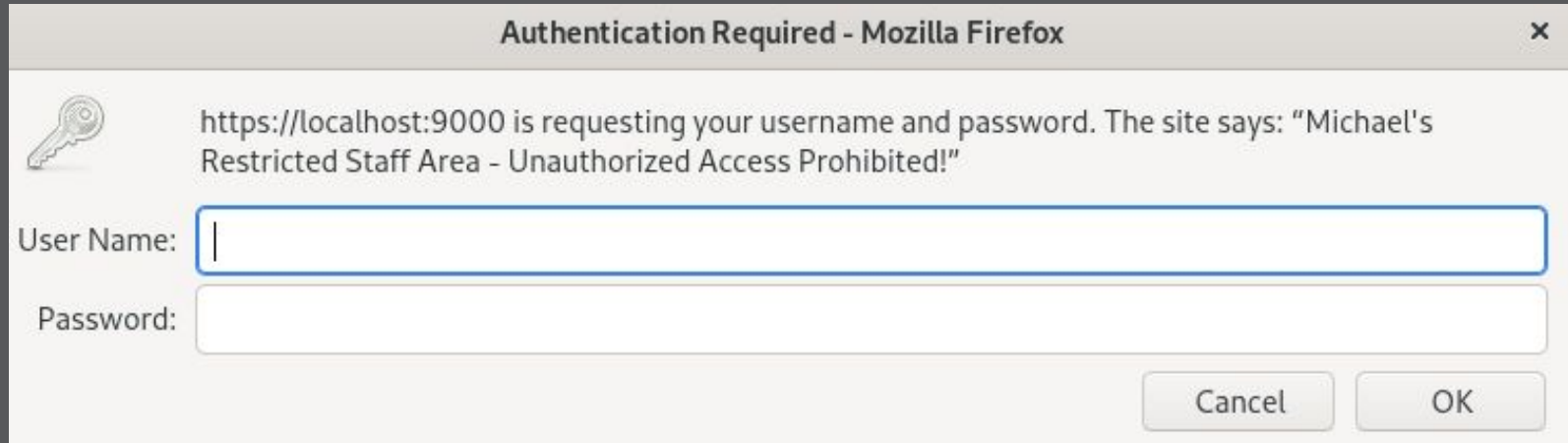
    <Directory /var/www/html/example.com/secret>
        AuthType Basic
        AuthName "Restricted Content"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>

</VirtualHost>
```

Enable apache mods, and disable other apache sites:


```
a2enmod rewrite  
a2enconf allow-override  
systemctl reload apache2  
a2dissite 000-default.conf  
systemctl restart apache2
```

Restart apache and browse to page:



The image shows a standard web browser authentication dialog box. It has a title bar that says "Authentication Required - Mozilla Firefox" with a close button (X) on the right. On the left side of the dialog, there is a small icon of a key. The main text area contains the message: "https://localhost:9000 is requesting your username and password. The site says: 'Michael's Restricted Staff Area - Unauthorized Access Prohibited!'". Below this message, there are two input fields. The first is labeled "User Name:" and is currently empty with a cursor. The second is labeled "Password:" and is also empty. At the bottom right of the dialog, there are two buttons: "Cancel" and "OK".

Authentication Required - Mozilla Firefox

 https://localhost:9000 is requesting your username and password. The site says: "Michael's Restricted Staff Area - Unauthorized Access Prohibited!"

User Name:

Password:

Cancel OK

Additional sites

michael.ferrie@edinburghcollege.ac.uk

Revision topic: be clear of the main steps in hosting a website with apache:

1) Make a directory for the website in the wwwroot:

```
mkdir -p /var/www/html/mysite.com
```

2) Set permissions and owner, create index file:

```
chown -R ec:ec /var/www/html/mysite.com
```

```
chmod -R 755 /var/www/html
```

```
cd /var/www/html/mysite.com
```

```
vi index.html
```

Additional Sites

3) Make the VirtualHost file for the website, set the DocumentRoot to the directory created in step 1:

```
cp default-ssl.conf mysite.com.conf  
    DocumentRoot var/www/html/mysite.com
```

4) Enable the site with a2ensite, reload:

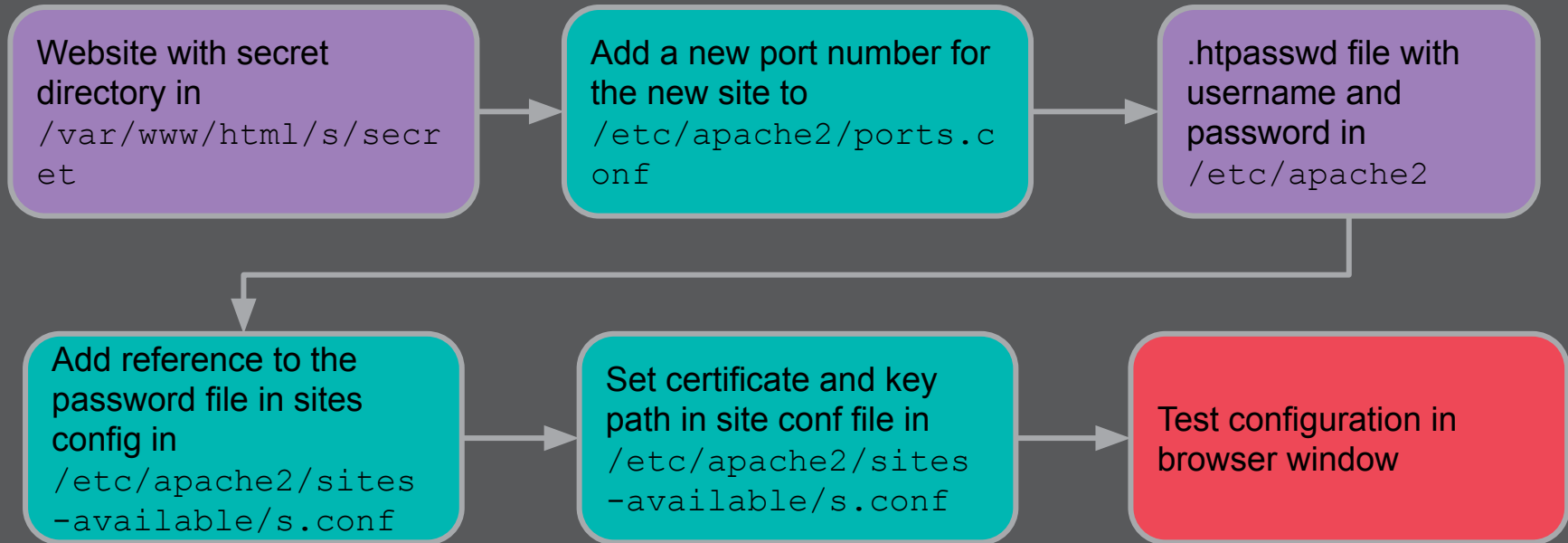
```
a2ensite mysite.com.conf
```

Additional Sites

- For https sites **only step 3** is different
- Copy the default-ssl.conf file
- Then specify a path in the VirtualHost file to a certificate and key, these can be created with the following openssl command:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/ssl/private/apache-my.key -out  
/etc/ssl/certs/apache-my.crt
```

.htaccess workflow



Key locations

```
/etc/apache2/apache2.conf  
/etc/apache2/sites-available  
/etc/apache2/sites-enabled  
/etc/apache2/ports.conf  
  
/var/www/html  
/var/log/apache2/access.log  
/var/log/apache2/error.log
```

Thanks for listening, any questions,
thoughts, ideas or reflections?

[Short video clip](#)

michael.ferrie@edinburghcollege.ac.uk