LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

Steps to secure apache using .htaccess and show how it can used to give secure access to the website

Student Name: Bhonagiri Sai Krishna

Student ID: 11403196

Email Address: skbsk232@gmail.com

GitHub Link:

Introduction

When running a website, there are often parts of the site that you'll want to restrict from visitors. Web applications may provide their own authentication and authorization methods, but the web server itself can also be used to restrict access if these are inadequate or unavailable. Apache lets you password protect individual files, folders, or your entire site easily.

This project shows how to password protected different web sites directories in Apache web server

How it works

To add password protection to your pages, you need to do the following two things:

1. Create a text file on your server that will store your username and password.
2. Create a special file called .htaccess in the folder you want to protect.

Step 1 — Installing the Apache Utilities Package

We will use a utility called htpasswd, part of the apache2-utils package, to create the file and manage the username and passwords needed to access restricted content.

- sudo apt-get update
- sudo apt-get install apache2-utils

Step 2 — Creating the Password File

We now have access to the htpasswd command. We can use this to create a password file that Apache can use to authenticate users. We will create a hidden file for this purpose called .htpasswd within our /etc/apache2 configuration directory.

The first time we use this utility, we need to add the -c option to create the specified file. We specify a username at the end of the command to create a new entry within the file:

- **sudo htpasswd -c /etc/apache2/.htpasswd user**

If we view the contents of the file, we can see the username and the encrypted password for each record:

- **cat /etc/apache2/.htpasswd**

Output:

user: $apr1$.0CAabqX$rb8lueIORA/p8UzGPYtGs/

Step 3 — Configuring Apache Password Authentication

Now that we have a file with users and passwords in a format that Apache can read, we need to configure Apache to check this file before serving our protected content. We can do this by placing .htaccess files in the directories that need restriction.

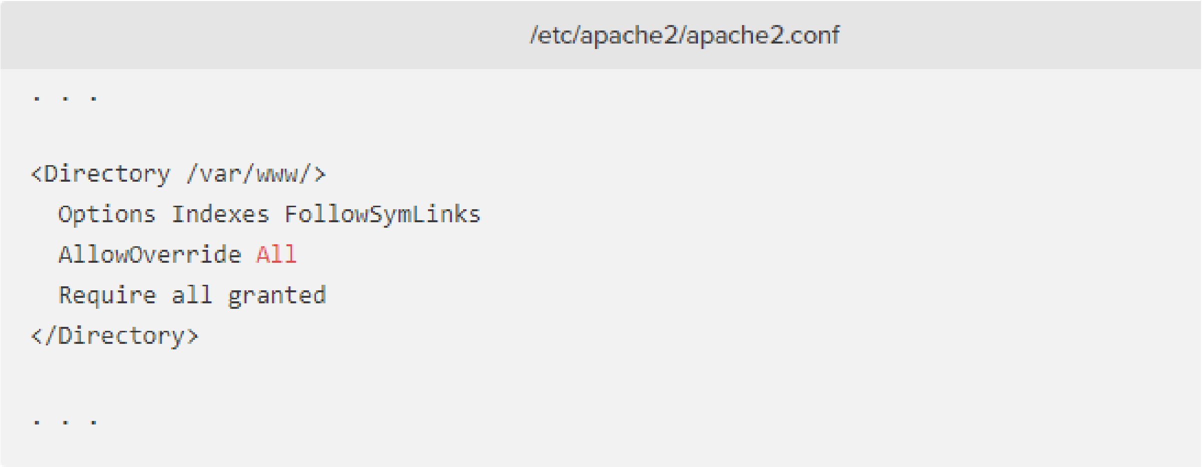
Configuring Access Control with .htaccess Files

Apache can use .htaccess files to allow certain configuration items to be set within a content directory. Since Apache must re-read these files on every request that involves the directory, using .htaccess file or need to allow non-root users to manage restrictions, .htaccess files make sense.

To enable password protection using .htaccess files, open the main Apache configuration file:

- **sudo nano /etc/apache2/apache2.conf**

Find the <Directory> block for the /var/www directory that holds the document root. Turn on .htaccess processing by changing the Allow Override directive within that block from "None" to "All":



/etc/apache2/apache2.conf

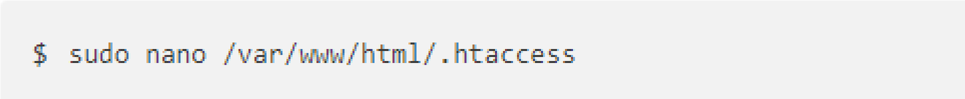
```
. . .

<Directory /var/www/>
  Options Indexes FollowSymLinks
  AllowOverride All
  Require all granted
</Directory>

. . .
```

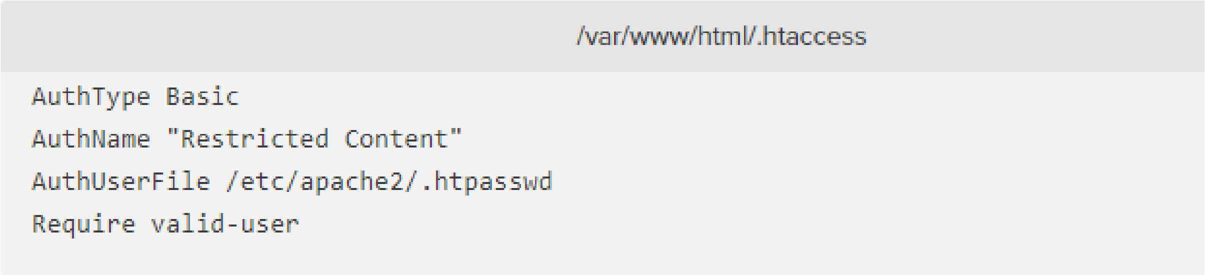
Save and close the file when you are finished.

Next, we need to add an .htaccess file to the directory we wish to restrict. In our demonstration, we'll restrict the entire document root (the entire website) which is based at /var/www/html, but you can place this file in any directory where you wish to restrict access:



```
$ sudo nano /var/www/html/.htaccess
```

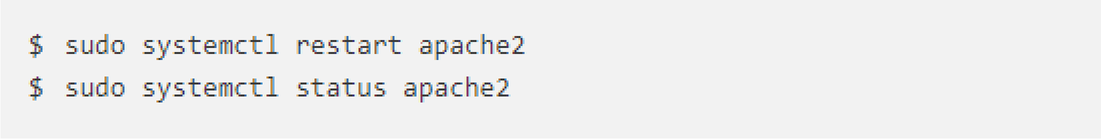
Within this file, specify that we wish to set up Basic authentication. For the AuthName, choose a realm name that will be displayed to the user when prompting for credentials. Use the AuthUserFile directive to point Apache to the password file we created. Finally, we will require a valid-user to access this resource, which means anyone who can verify their identity with a password will be allowed in:



/var/www/html/.htaccess

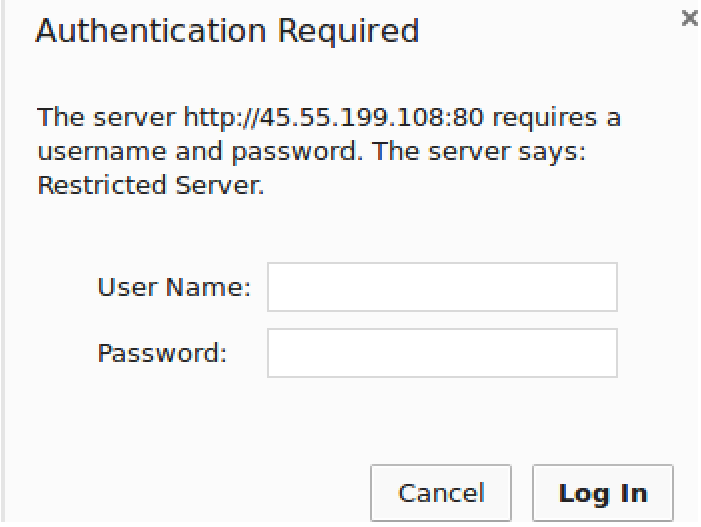
```
AuthType Basic
AuthName "Restricted Content"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
```

Save and close the file. Restart the web server to password protect all content in or below the directory with the .htaccess file and use systemctl status to verify the success of the restart:



```
$ sudo systemctl restart apache2
$ sudo systemctl status apache2
```

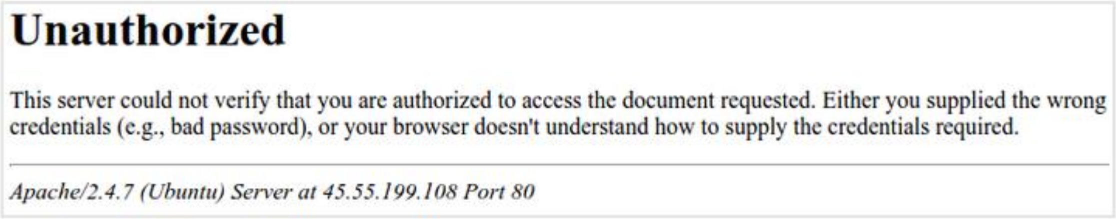
To confirm that your content is protected, try to access your restricted content in a web browser. You should be presented with a username and password prompt that looks like this:



Authentication Required

The server http://45.55.199.108:80 requires a username and password. The server says: Restricted Server.

If you enter the correct credentials, you will be allowed to access the content. If you enter the wrong credentials or hit "Cancel", you will see the "Unauthorized" error page:



Unauthorized

This server could not verify that you are authorized to access the document requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply the credentials required.

Apache/2.4.7 (Ubuntu) Server at 45.55.199.108 Port 80

Conclusion

We have now set up basic authentication for our site. Apache configuration and .htaccess can do much more than basic authentication.