By Nayyar Ahmad
Published: 2007-01-22 16:34

.htaccess Based Authentication On Subdirectories

Author: Nayyar Ahmad.

Contact: nayyares *AT* gmail *DOT* com

nayyar.ahmad *AT* cilix *DOT* co *DOT* mz

Dedication: ***To Baji, for all those efforts, she did for me.***

Preface:

.htaccess is used to provide facility of changing configuration per directory basis, this file can contain one or more directives that are going to be forced on the directory that contains the.htaccess file.

`.htaccess` is a popular name of this configuration modification file, you can change this default name by changing *`AccessFileName`* parameter in Apache config file. i.e.

```
AccessFileName .anyname
```

In this howto, I will take a dummy scenario of two parallel level subdirectories and will implement password authentication on both of them. This means only those users can access these directories that have the correct username and password.

So it begins :)

Here I am assuming that your DocumentRoot directory is *`/var/www/html`* but if you have VirtualHost configuration or even Apache is configured on some other root directory then you can adjust this according to your situation.

Note: I will take two dummy names for these directories as `test-dir1` and `test-dir2`.

1 Creating Directory:

```
$ cd  /var/www/html
```

```
$ mkdir  test-dir1
```

```
$ mkdir test-dir2
```

2 Test HTML File Creation:

Creating html file in first directory.

```
$ cd  /var/www/html/test-dir1
```

```
$ cat > index.htm
```

```
<html>

<head> TEST PAGE IN TEST-DIR1. </head>

</html>
```

Creating html file in second directory.

```
$ cd /var/ www/html/test-dir2
```

```
$ cat  > index.htm
```

```
<html>

<head> TEST PAGE IN TEST-DIR2.</head>

</html>
```

3 Browsing Test Pages:

Now you can browse and test, whether the pages are available or not, by opening any web browser, I like Firefox (as it rocks :)) and type:

`http://localhost/test-dir1/`

This will display the first directory test page, and

`http://localhost/test-dir2/`

will display the second test page in *`test-dir2`*.

If you are able to see both pages, it means that you are almost about to rock .htaccess.

4 .htaccess File Creation:

```
$ cd  /var/www/html/test-dir1
$ vi  .htaccess
```

Write the following lines into this file:

```
AuthName "Authorized Users Only."

AuthType Basic

AuthUserFile /etc/httpd/conf/.htpasswd

require user testusr
```

Now I will explain, what magic lines we have written in this file:

AuthName parameter just defines the title of the password entry box when the user logs in, while the *AuthType* tells the server what sort of processing is in use, and *Basic* is the most common and perfectly adequate for almost anypurpose. *AuthUserFile* is used to define the *.htpasswd* file location, this files contains the password of the user who is going to be authenticate in *.htaccess* file. *require user* is used to identify the trusted user, if there are more than one trusted user, then you can specify their names in a space saparated list.

Now to make *test-dir2* protected by *.htaccess*, we need to copy it from *test-dir1* to *test-dir2* with the following command:

```
$ cp /var/www/html/test-dir1/.htaccess   /var/www/html/test-dir2/
```

5 User Creation:

Here we will create a test user to check our *.htaccess* :)

```
$ adduser   testusr
$ passwd   testusr
```

6 Telling Apache About Users:

Now we have to inform Apache about the user and its password, but before going into this step there is a social duty on me i.e. to explain both RPM and source Apache installation difference. :) If you have installed Apache from RPM then it will install all related commands in your `/usr/local/bin`, so no problems, you can give `htpasswd` command anywhere in your system, but if you have installed Apache from source then you have to find the Apache bin directory to execute the `htpasswd` command. In this HowTo I will give both ways, here it is:

```
$ htpasswd  -c /etc/httpd/conf/.htpasswd  testusr
```

The above command will work if you have `htpasswd` in your `/usr/local/bin` and it happens if you install Apache from RPM. `/etc/httpd/conf/.htpasswd` is the location of file that will contain the authenticated/trusted user password.

OR

```
$ cd /apache/bin/
```

```
$ ./htpasswd -c /etc/httpd/conf/.htpasswd   testusr
```

The above commands correct if you have installed Apache from the sources, `$ cd /apache/bin` can be adjusted according to your system, as maybe you have installed it somewhere else.

7 .htpasswd File Permission:

We need to set the file permission of the `.htpasswd` file and make the `apache` user the owner of this file.

```
$ chown apache.apache  /etc/httpd/conf/.htpasswd
```

8 Editing httpd.conf:

Now we have to edit the `httpd.conf`, as Apache needs to be informed about `.htaccess`, here we will change `AllowOverride All | none` to `Authconfig` , now there are two cases, one if you are hosting just one site and other if you are having VirtualHost, here is the First Case:

In this case you, we have only one `Directory` tag in `httpd.conf` file as we are hosting just one site, so we will edit the `<Directory>` tag for `/var/www/html`.

```
<Directory "/var/www/html">
  AllowOverride AuthConfig
  Order allow,deny
  Allow from all
</Directory>
```

Now for second case, when we have several sites hosted, i.e. VirtualHost:

```
<VirtualHost www.cbtcandy.org>
  DocumentRoot /var/www/html/cbtcandy
  ServerName www.cbtcandy.org
  <Directory /var/www/html/cbtcany>

    AllowOverride AuthConfig
    Order allow,deny
    Allow from all
    Options -Indexes
  </Directory>
</VirtualHost>
```

9 Restarting Apache:

Now you have to restart the Apache server to reload the configuration.

For RPM based system:

```
$ service httpd restart
```

For source based system, adjust your Apache's bin directory path.

```
$ /apache/bin/apachectl restart
```

10 Testing:

Now everything is ready to be tested, again open your favourite browser and try to open the following links:

http://localhost/test-dir1/

and

http://localhost/test-dir2/

Note: When you browse these linksyou will be asked for the username and password, once you provide them it will take you to the test page. But once you log in to one directory it will not require the username and password for the other test directory, as Apache will not ask for the username and password again and again for directories equal in level or subdirectories. So once you are authenticated the child and parallel directories are open to use. But if you still want to check them then use `links` ***text based browser, that is what I do for checking them.***