**HTACCESS in Apache**

* The **.htaccess** file is a hidden text file that allows you to control the way visitors are able to access your site. Some common uses of the **.htaccess** file would be restricting access to certain files, or redirecting URLs.
* Using .htaccess files lets you control the behavior of your site or a specific directory on your site. For example, if you place an .htaccess file in your root directory, it will affect your entire site (www.coolexample.com). If you place it in a /content directory, it will only affect that directory ([www.coolexample.com/content](http://www.coolexample.com/content)).

Using an .htaccess file, you can:

* Customize the Error pages for your site.
* Protect your site with a password.
* Enable server-side includes.
* Deny access to your site based on IP.
* Change your default directory page (index.html).
* Redirect visitors to another page.
* Prevent directory listing.
* Add MIME types.

It is used to create the password protected directories under Apache web server.

1. Stop httpd and iptables services at initial steps.

[root@sunil ~]#service iptables stop

[root@sunil ~]#service httpd stop

2. Make sure Apache is configured to use .htaccess file: Open httpd.conf file and check for Directory entry as shown in below and save the given changes.

[root@sunil ~]#vi /etc/httpd/conf/httpd.conf

<Directory "/var/www/icons">

Options Indexes MultiViews

AllowOverride None

Order allow,deny

Allow from all

</Directory>

Replace the above directory entry by the following entry having "DocumentRoot" /var/www/html/

<Directory /var/www/html>

Options Indexes Includes FollowSymLinks MultiViews

AllowOverrideAuthConfig

Order allow,deny

Allow from all

</Directory>

3.Create a password file with htpasswd command and follow the instructions in 3.1,3.2,3.3

General syntax: **htpasswd-c password-fileuser\_name**

Where,

\* -c: Create the password-file. If password-file already exists, it is rewritten and truncated.

\* username: The username to create or update in password-file. If username does not exist

in this file, an entry is added. If it does exist, the password is changed.

3.1.Create any directory where we can save the password file outside apache document root, so that only Apache can access the password file. This is so that people cannot download the password file from the web site/web page. Create an empty file in that directory having any name say “.htpasswd”. Here we had given “.” before the file to make it hidden.

[root@sunil ~]# mkdir -p /usr/sunil/passwd

[root@sunil ~]#touch/usr/sunil/passwd/.htpasswd

3.2.i)Now you are able to add new user called nikhil. We can add multiple users without -c option as shown below:

[root@sunil ~]# htpasswd –c /usr/sunil/passwd/.htpasswd nikhil

New password:

Re-type new password:

Adding password for user Nikhil

[root@sunil ~]# htpasswd /usr/sunil/passwd/.htpasswd nilesh

New password:

Re-type new password:

Adding password for user nilesh

3.2.ii) If you want to modify the password of existing user then use the following command:

[root@sunil ~]#htpasswd /usr/sunil/passwd/.htpasswd name-of-user

3.3.Then check for the "apache" user and group in "httpd.conf", "/etc/passwd" and "/etc/group" as shown below. Apache httpdautomatically creates an apache as default user and group.

[root@sunil ~]# grep apache /etc/passwd

[root@sunil ~]# grep apache /etc/group

[root@sunil ~]#grep apache /etc/httpd/conf/httpd.conf

User apache

Group apache

3.4. Now change the user, group and permissions of password file.You have to give the minimum permissions to the password file.

# chownapache:apache /usr/sunil/passwd/.htpasswd

# chmod 660 /usr/sunil/passwd/.htpasswd

3.5. You can check the owner, group and permissions of the ".htpasswd" file using:

[root@sunil ~]# ls –lta /usr/sunil/passwd/.htpasswd

-rw-rw---- 1 apache apache 21 Aug 9 14:36 .htpasswd

3.6.You can check the content of password file. Here the users are readable but the password is in the encrypted format.

[root@sunil ~]# cat /usr/sunil/passwd/.htpasswd

nikhil:p1aBayukZFpbE

nilesh:piyuREfNBHGaa

4. Let us assume you have directory called “/var/www/html/aaa” and you would like to protect it with a password. Create a directory /var/www/html/aaa if it does not exist:

[root@sunil ~]# # mkdir -p /var/www/html/aaa

Create .htaccess file using text editor:  
[root@sunil ~]#vi /var/www/html/aaa/.htaccess

Add following text in the .htaccess file and save the given text in it:

AuthType Basic

AuthName "Restricted Access"

AuthUserFile /usr/sunil/passwd/.htpasswd

Require user nikhilnilesh

# service iptables start

# service httpd start

5. Test your configuration: You can test from the local or remote machine by simply entering the URL http://localhost/aaa for local machine and http://IP\_Address\_Server\_Machine/aaa for local or remote system. Then it will pop-up the window asking for “User Name” and “Password”. Enter the valid user name (either nikhil or nilesh) and the password for the respective users. If either user name or password gets wrong then it shows the message authorization failed.

6. There is one more problem with it is that when we run the web page for first time then it

will ask for user name and password but when you requested again the it will not ask for

username and password again(Even after pressing the Shift+Refresh button). This is

because the web page is get cached and we need to delete the cached. Avoid this problem

using two ways

i) Close the given browser. Reopen it and request again for the same web page

ii) Instaed of closing the browser clear the "Authenticated Sessions" which is inside browser" edit-->Preferences-->Privacy-->Clear now-->Authenticated Sessions" and clear

that data and again refresh the web page then it will ask for username and password.

7. One more point to remember is that you can add following lines to file <Diretory> entry in httpd.conf file. And To change or setup new user use htpasswd command again. It is an optional step either use .htaccess file or the following entry in .conf file.

AuthType Basic

AuthName "Restricted Access"

AuthUserFile /usr/sunil/passwd/.htpasswd

Require user nikhil nilesh

**Adding Groups in the .htaccess file:**

1. Add multiple users in a .htpasswd file using “htpasswd” command by giving the path of password file “.htpasswd”
2. Create a group file in the directory where we put the .htpasswd fileSuppose we have the .htpasswd file in the directory "/usr/sunil/passwd” then create the user group file named as ".htgroup" in the same directory

[root@sunil ~]#touch /usr/sunil/passwd/.htgroup

1. Change the "ownership" and "permission" of “.htgroup” file same as that of .htpasswd file as shown

-rw-rw---- 1 apache apache13 Aug 12 14:32 .htgroup

-rw-rw---- 1 apache apache 168 Aug 12 14:11 .htpasswd

1. Open the ".htgroup" file and make the entry of groups and users in it.The file may have single/multiple users/groups. You can also add the single user in multiple groups. A group file consists of lines giving a group name followed by colon and a blank space and list of users separated by blank space in that group. For example:

group\_name1: user1 user2 user3

group\_name2: user1 user4 user5

1. Now open the ".htaccess" file in the directory to which we have to make it a password protected for a perticular groups/users and make the following entries in it. You can make user entry in “Require” directive but it will not properly work because it is treated as a filtering criteria mentioned in the following note given.

AuthType Basic

AuthName "Restricted Resource"

AuthUserFile /web/users

AuthGroupFile /web/groups

Require group admin

**Note:** When multiple Require directives are used in a single .htaccess file, then it check for the first require directive and is true then it goes for second require directive otherwise the page is not accessible. It is similar to if inside theif statement in C language.

**Thus the first one to authorize a user authorizes the entire request, and subsequent “Require” directives are ignored. In the given example only the "admin" group is able to access the web page.**

There's another way to let multiple users in that is less specific. Rather than creating a group file, you can just use the following directive:

**Require valid-user**

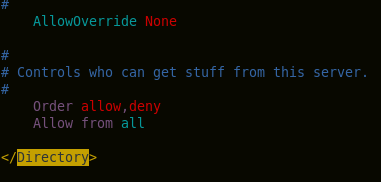
Using this statement rather than the “Require user user\_name1” line multiple times it allows anyone user in the password file is allowed to access this web pagewho correctly enters their valid user name and password.

**Allowing and Restricting the users by IP Addresses:**

For this purpose we make the entry either into the "httpd.conf" file or ".htaccess" file. Here we search for "order allow,deny" clause in .conf file. This is obtained where we already make the changes for running the htaccess.Make the following entries after the "order allow,deny" clause where we allow/deny from selected IP addresses to get access the web pages. You can also able to "allow from all" or "deny from all" except some IP address you want to give some permissions.

allow from 192.168.1.2

deny from 192.168.1.5



To run the web page restart the service and run the web page.

**Note:** Do NOT put a space before or after the comma in "allow,deny". The server will consider this a syntax error and will not allow anyone to access your directory.