PasswordBasicAuth

Password protect a directory using basic authentication

In this How-To guide, we will show you how to set up a password protected directory using basic authentication.

The first section focuses on Apache httpd 2.2, and the new directives for 2.4 will be covered in the last part of this document.

Authentication directives in Apache httpd can be used in the following contexts - directory and htaccess. For directory context this means in **<Directory>**, **<Location>**, and **<Files>** blocks in your httpd.conf or your distro's main Apache config file or virtual host config file. Additionally, for Apache httpd 2.2, **<Proxy>** blocks are also included in the directory context. The htaccess context is self explanatory. This means you can use authentication directives in .htaccess files. In this tutorial, we will show recipes for both contexts.

The first thing we need to do in this example is to create a directory to protect in our document root. Let's say our document root is **/var/www/html**. We'll create a directory called **protected** in the document root - **/var/www/html/protected**.

The next thing to do is to create a password file with users. We will use the htpasswd utility provided in the core Apache package. The password file can be stored anywhere on your hard drive. In our example we will create our htpasswd file in **/etc/htpasswd**.

*Note that the location of the htpasswd file can be anywhere you want on your local drive. You just need to specify the full path to the htpasswd file with the **AuthUserFile** directive. Choose whatever you deem to be a sane location for your password files.*

```
/path/to/htpasswd -c /etc/htpasswd/.htpasswd user1
/path/to/htpasswd /etc/htpasswd/.htpasswd user2
```

/path/to/ is the full path to the htpasswd utility. The full path to the htpasswd utility is necessary if htpasswd is in a nonstandard location. After running the htpasswd command, you will be prompted to enter the user's password. Notice the difference between both commands. The first command uses the **-c** flag. This flag is used when creating a new htpasswd file. After that, the **-c** flag is not used for subsequent users you wish to add. Also, you need to make sure Apache has read access to this file, so make sure your permissions are correct.

This is the recipe to use for setting up a password protected directory in the directory context:

```
<Directory "/var/www/html/protected">
  AuthType Basic
  AuthName "Authentication Required"
  AuthUserFile "/etc/htpasswd/.htpasswd"
  Require valid-user

  Order allow,deny
  Allow from all
</Directory>
```

The lines to focus on are **AuthType**, **AuthName**, **AuthUserFile**, and **!Require**.

1. AuthType tells Apache what type of authentication to use. In our case, basic authentication.
1. AuthName is what will be displayed on the password prompt from the browser.
1. AuthUserFile is the location of your htpasswd file.
1. Require tells Apache which authenticated users will be granted access to a resource. In our case, any authenticated user will be granted access.

The following below is the recipe to use for setting up a password protected directory in the htaccess context:

First we will create a .htaccess file in our protected directory, **/var/www/html/protected** and set the contents of the file to be:

```
AuthType Basic
AuthName "Authentication Required"
AuthUserFile "/etc/htpasswd/.htpasswd"
Require valid-user
```

Now we need to create a **<Directory>** block in httpd.conf or your distro's main apache config file or your virtual host config file in order to have Apache process this htaccess file.

```
<Directory "/var/www/html/protected">
  AllowOverride AuthConfig
  # The Options below is an example. Use what you deem is necessary.
  Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
  Order allow,deny
  Allow from all
</Directory>
```

Notice the **AllowOverride** line. It tells Apache to process the htaccess file and to allow htaccess to set the authentication for that directory.

Remember to restart Apache after making any changes to httpd.conf or your distro's main Apache config file or your virtual host config file.

Using either recipe, you can now go to http://localhost/protected and be prompted by the browser to enter your credentials. If you enter correct credentials you will be granted access to **protected**. If you don't enter correct credentials, you will be continually prompted to enter credentials until you enter correct credentials or click the **Cancel** button.

For Apache httpd 2.4, the authorization mechanism has been revamped. Here is a sample of a configuration that uses basic HTTP auth for the entire DocumentRoot, and allows public, non-restricted access for a specific directory:

Assuming a DocumentRoot value of "/srv/httpd/htdocs",

```
<Directory "/srv/httpd/htdocs">
    Options +FollowSymLinks +Multiviews +Indexes
    AllowOverride None
    AuthType basic
    AuthName "private"
    AuthUserFile /srv/httpd/.htpasswd
    Require valid-user
</Directory>

<Directory /srv/httpd/htdocs/public>
    Require all granted
</Directory>
```

The password file would be created in the same fashion as it would be on 2.2.

For more complete information on the Apache directives used, see the Apache Docs.