Change website .JS .PHP Python Perl CSS etc. file permissions recursively for Better Tightened Security on Linux Webhosting Servers

Author : admin

It is a common security (breach) mistake that developers or a web design studio make with dedicated or shared hosted websites do to forget to set a nice restrictive file permissions.

This is so because most people (and especially nowdays) developers are not a security freaks and the important think for a programmer is to make the result running in shortest time without much caring on how secure that is.
Permissions issues are common among sites written in **PHP / Perl / Python with some CSS and Javascript**, but **my observations are that JavaScript websites especially that are using some frameworks such as Zend / Smarty etc. and are using JQuery are the most susceptible to suffer from permission security holes** such as the **classic 777 file permissions**, because of developers who're overworking and pushed up for a deadlines to include new functionality on websites and thus often publish their experimental code on a Production systems without a serious testing by directly uploading the experimental code via FTP / WinSCP on Production system.

Such scenarios are very common for *small and middle sized companies websites* as well as many of the hobbyist developers websites running on ready CMS system platforms such as Joomla and Wordpress.
I know pretty well from experience this is so. Often a lot of the servers where websites are hosted are just share-servers without a dedicated sysadmin and thus there are no routine security audits made on the server and the security permissions issue might lead to a serious website compromise by a cracker and make your website quickly be banned from **Google / Yahoo / Ask Jeeves / Yandex** and virtually most of Search Engines because of being marked as a spammer or hacked webiste inside some of the multiple website blacklists available nowdays.

Thus it is always a good idea to keep your server files (especially if you're sysadmin) with restrictive permissions by making the files be owned by superuser (root) in order to prevent some XSS or vulnerable PHP / Python / Perl script to allow you to easily (inject) and overwrite code on your website.

1. Checking whether you have a all users read, write, executable permissions with find command

The first thing to do on your server to assure you don't have a low security permissioend files is:

```
find /home/user/website -type f -perm 777 -print
```

You will get some file as an output like:

```
./www/tpl/images/js/ajax-dynamic-list/js/ajax-dynamic-list.js
./www/tpl/images/js/ajax-dynamic-list/js/ajax_admin.js
./www/tpl/images/js/ajax-dynamic-list/js/ajax_teams.js
./www/tpl/images/js/ajax-dynamic-list/js/ajax.js
./www/tpl/images/js/ajax-dynamic-list/js/ajax-dynamic-list_admin.js
./www/tpl/images/js/ajax-dynamic-list/lgpl.txt
```

2. Change permissions recursively to read, write and exec for root and read for everybody and set all files to be owned by (root) superuser

Then to fix the messy permissions files a common recommended permissions is ***744*** **(e.g. Read / Write and Execute permissions for everyone and only read permissions for All Users and All groups)**. Lets say you want to make files permissions to 744 just for all JavaScript (JQuery) files for a website, here is how:

```
find . -iname '*.js' -type f -print -exec chown root:root '{}' \;
find . -iname '*.js' -type f -print -exec chmod 744 '{}' \;
```

First find makes all Javascript files be owned by root user / group and second one sets all files permissions to 744.

To make 744 all files on server (including JPEG / PNG Pictures) etc.:

```
find . -iname /home/users/website -type f -print -exec chown root:root '{}' \;
find . -iname /home/users/website -type f -print -exec chmod 744 '{}' \;
```