Adding another level of security to your shared Debian Linux webhosting server with SuPHP

Author : admin

There are plenty of security schemes and strategies you can implement if you're a **Shared Web Hosting company sysadmin** however probably the most vital one is to install on **Apache + PHP Webserver SuPHP module**.

 # apt-cache show suphp-common|grep -i descrip -A 4

Description: Common files for mod suphp Suphp consists of an Apache module (mod_suphp for either Apache 1.3.x or Apache 2.x) and a setuid root binary (suphp) that is called by the Apache module to change the uid of the process executing the PHP interpreter to the owner of the php script.

So **what SuPHP actuall does** is to *run separate* **CPanel / Kloxo** etc. Users with separate username and groupid permissions coinciding with the user present in **/etc/passwd** , **/etc/shadow** files existing users, thus in case if someone *hacks some of the many customer sites he would be able to only write files and directories under the user with which the security breach occured*.

On servers where *SuPHP* is not installed, all systemusers are using the same **UserID / GuID** to **run PHP executable scripts under separate domains Virtualhost** which are coinciding with *Apache* (on **Debian / Ubuntu** *uid, gid* - **www-data**) or on (**CentOS / RHEL / Fedora** etc. - user **apache**) so once one site is defaced exploited by a worm all or **most server websites might end up infected with a Web Virus / Worm which will be trying to exploit even more sites of a type running silently in the background.** This is very common scenarios as currently there are donezs of **PHP / CSS / Javasripts / XSS vulnerability exploited on VPS and Shared hosting servers** due to *failure of a customer to update his own CMS scripts / **Website (Joomla, Wordpress, Drupal** etc.)* and the lack of resource to regularly

monitor all customer activities / websites.

Therefore installing **SuPHP Apache module** is *essential one to install on new serverslarge hosting providers as it saves the admin a lot of headache from spreading malware across all hosted servers sites* ..
Some ***VPS*** *admins* that are security freaks tend to also install **SuPHP** module together with many chrooted ***Apache / LiteSpeed / Nginx webservers*** each of which running in a separate Jailed environment.

Of course using *SuPHP besides* giving a improved security layer to the webserver has its downsides such as increased load for the server and making ***Apache PHP scripts*** *being interpretted a little bit slower than with plain* ***Apache + PHP*** but performance difference while running a site on top of *SuPHP* is often not so drastic so you can live it up ..

Installing SuPHP on a Debian / Ubuntu servers is a piece of cake, just run the as *root superuser*, usual:

```
# apt-get install libapache2-mod-suphp
```

Once installed only thing to make is to **turn off default installed Apache PHP module** (without SuPHP compiled support and restart Apache webserver):

```
# a2dismod php5 ...
```

```
# /etc/init.d/apache2 restart
...
```

To test the SuPHP is properly working on the Apache Webserver go into some of many hosted server websites **DocumentRoot**

And create new file called test_suphp.php with below content:

```
# vim test_suphp.php
```

Then open in browser ***http://whatever-website/test_suphp.php*** assuming that system(); function is not disabled for security reasons in php.ini *you should get an **User ID, GroupID** bigger than reserved system IDs on GNU / Linux e.g. **ID > UID / GID 99***

Its also a good idea to take a look into SuPHP configuration file **/etc/suphp/suphp.conf** and tailor options according to your liking

If different hosted client users home directories are into **/home** directory, set in **suphp.conf**

```
;Path all scripts have to be in

docroot=/home/
```

Also usually it is a good idea to set

```
umask=0022
```