Generating Static Source Code Auditing reports with Spike PHP Security Audit Tool

Author : admin

I'm conducting a PHP Audit on a server in relation to that one of the audit criterias I follow is a **Static PHP Source Code Auditing** of the php files source code located physically on the Linux server. Auditing a tons of source code manually is a kind of impossible task, therefore I needed a quick way to at least
partly automate or fully automate the PHP applications source code.
A quick search in Google pointed me to a php application tool - Spike Security Audit .
This small application PHP written app is quite handy. It is able to either check a certain php source code file for **WARNINGS** or **ERRORS** or do a complete security source code analysis of a bunch of PHP files in a directory including all the other php source files in subdirectories.

After executed the PHP Security Audit Tool generates a nice source code analysis report in html that can easily be later observed with some Browser.

The use of the tool is pretty straight forward, all you have to do is download it from Spikeforge - the project's official webpage and unzip it e.g.

```
debian-server:~# wget
http://developer.spikesource.com/frs/download.php/136/spike_phpSecAudit_0.27.zip
debian-server:~# unzip spike_phpSecAudit_0.27.zip
```

Then you have to invoke the run.php with the php cli, that you need to have installed first.
If you don't have the php cli yet please install it with the command:

```
debian-server:~# apt-get install php5-cli
```

Now you have to execute the **run.php** script bundled with the spike php security audit program source code.

```
debian-server:~# php run.php

Please specify a source directory/file using --src option.

Usage run.php options

Options:
--src Root of the source directory tree or a file.
--exclude [Optional] A directory or file that needs to be excluded.
--format [Optional] Output format (html/text). Defaults to 'html'.
--outdir [Optional] Report Directory. Defaults to './style-report'.
--help Display this usage information.
```

As you can see the spike php security audit has only few command line options and they're quite easily understandable.
However in my case I had to audit a couple of directories which contained source code.
I also wanted the generated reports to be cyclic, on let's say per daily basis cause I wanted to have the PHP applicaiton analysis generated on a daily basis.
In that reason I decided to write a small shell script that would aid the usage of php spike audit, I've called the script code-analysis.sh

The usage of the Automation source code analysis script for PHP Spike Audit can be downloaded here
The script has a few configuration options that you might need to modify before you can put it to execute on a crontab.

This are:

# Specify your domain name on which php spike audit reports will be accessed
domain_name='yourdomainname.com';
# put here the location where phpspike run.php execute is located
spike_phpsec=/usr/local/spike_phpSecAudit_0.27/run.php;
# specify here which will be the directory where the php source code analysis reports will be stored by php spike
log_dir=/root/code-analysis/;
# in that part you have to specify the physical location of the php cli it's located by default in /usr/bin/php on Debian GNU Linux.
php_bin=/usr/bin/php;
# the directory below should be set to a directory where the reports that will be visible from the webserver will be stored
www_dir=/var/www/code-analysis;

in the variables

directory[1]='/home/source-code1/'; ..
directory[2]=''; ..

you should configure the directories containing php source code to be audited by the php spike audit tool.

After you have prepared the code-analysis.sh script with your custom likings, you can now put it to be executed periodically
using crontab or some other unix system scheduler of choice.

To do that edit your root crontab.

crontab -u root -e

and put in it.

```
# code analysis results
05 3 * * * /usr/local/bin/code-analysis.sh >/dev/null 2>&1
```

Now hopefully you can edit your **/etc/apache2/apache2.conf** or your **httpd.conf** depending on your linux or unix architecture and make a **Alias** like:

```
Alias /code-analysis "/var/www/code-analysis"
```

Now your php source code analysis from the php spike audit tool will be generated daily.
You will be able to access them via web using http://yourdomain.com/code-analysis/

That way, you can review your php source code written or changed in your php applications on daily basis and you can a way easily track your coding mistakes, as well as track for possible security issues in your code.

For the sake of security I've also decided to protect the /code-analysis Apache directory with a password using the following .htaccess file:

```
AuthUserFile /var/www/code-analysis/.htpasswd
AuthGroupFile /dev/null
AuthName "Login to access PHP Source Code Analysis"
AuthType Basic

require valid-user
```

If you decide to protect yours as well you have to also generate the .htpasswd file using the following command:

```
debian-server:~# htpasswd -c /var/www/code-analysis/.htpasswd admin
```

You will be asked for a password. The code-analysis.sh script will also take care to generate an html file for you including links to reports to all the php source code audited directories reports.

Now accessing **http://yourdomain.com/code-analysis/** will give you shiny look to the php source applications generated reports .