## Permessi apache : (x aggiungere pagine nuove)

chown -R www-data:www-data /var/www/mysite

chmod -R og-r /var/www/mysite

## shell\_exec con sudo:

<?php

$outPut = shell\_exec("echo password\_for\_the\_user | sudo -S command\_to\_execute");

echo "<pre>$outPut</pre>";

?>

## Oppure: (facendo uno script)

www-data ALL=NOPASSWD: /path/to/script

NB.

Giving the user PHP runs as root rights is **highly dangerous**. This is not a good idea at all, because you open up your whole server if a vulnerability in a PHP script gets exploited. On a production server, this is absolutely not acceptable.

The only way to do this securely that I know of is having a cron job run a shell script as root every minute or so. The shell script tests for the presence of a file like shutdown\_now.txt. If the file exists, the script starts the shutdown procedure. The PHP script creates the shutdown file if so instructed.

Code:

if ($\_POST['action']=="Shutdown") {

$test=shell\_exec("touch /tmp/shutdown");

}

Shell script /usr/sbin/check\_shutdown

Code:

#!/bin/sh

if [ -f "/tmp/shutdown" ]

then

rm -f /tmp/shutdown

/sbin/shutdown now

fi

Edit root's crontab using "sudo crontab -e" and add the line

Code:

\* \* \* \* \* sh /usr/sbin/check\_shutdown

This runs the check\_shutdown script once each minute. If it finds the /tmp/shutdown file created by the PHP script, it executes the shutdown command.