Blocking Automated Submissions (capatcha) php script

Attachment	Size
blockautosubmit_v0.2.tar.gz [1]	652.41 KB
blockautosubmit_v0.1.tar.gz [2]	292.43 KB

A php script freely available to give back to the open source [3] community

This script is no longer actively supported but due to requests we put it back up here for historical reference and it provides an example from 2003 of using php for a capatcha. The last time it was touched was back in 2006. But we do know that modifications of the script are still being used for special purpose websites as of 2012.

NOTE: Since this script was made available awhile back, several other projects with similar purpose have become available and one derived from this. The naming that has become most frequently used for this purpose is CAPTCHA [4], *c*ompletely *a*utomated *p*ublic *T*uring test [5] to tell *c*omputers and *h*umans *a*part

These scripts can be defeated by using OCR. If you are a company protecting finances or privacy, spend the resources to do something better and hire some writers and make it open source to make it safer with community oversight. We include the following information and a working Open Source example of applications that adequately work, provide a good starting place for concept, but could be defeated with someone investing labor and resources to setup an attack with OCR recognition.

See Also the Wikipeida definition of Captcha that give more information
http://en.wikipedia.org/wiki/Captcha [4]

-

    p.s. You may also want to set a date/time check on your mail delivery if cc goes to a pager, to disable messaging during the 6 hours you get to sleep as an sysadmin..

- **A derivation with improvements has been brought to our attention:**
    http://www.activistinabox.org/blockautosubmit/ [6]

Blocking Automated Submissions using image features of php

A folder with php code to require viewer to identify characters in an image before form submission is allowed. Features index.php storing the random characters in session for generateimage.php to use, some settings in config.php, with a unique twist it searches a subfolder 'ttf' for true type fonts to use while composing the characters on the image so you can add or remove fonts on the fly. Do be careful about copyright on true type fonts!

Requirements:

- PHP [7] supporting version GD [8] 1.6.3 or better (may or may not work lower than that, tested with gd-1.8.3). Effort was made to avoid php functions requiring GD 2.x or later however that may impact using more colors and

possibly type 2 true type fonts. Let us know your experiences **[9].**
  - **True Type Font rendering for** GD **[8] program, tested with** freetype **[10]-1.3.1**
  - **And probably several other applications that were not obvious to you, let us know what was not obvious and we will list here :)**

License was intended to be BSD style so that it can be used and modified by others for commercial and non-commercial use
[11]

FAQ

Q. The image appears but the characters are not showing up, suggestions?

A. If for some reason the session is not registering the random character string created on the form page then when the web server tries to open the graphic "generateimage.php" that php script won't be able to get the string from the session so you may want to verify that the scripts are able to use the session properly to store the important random string for the image.
Test that your setup with php can create images with text using the functions like is_dir and imagegettftext that are used in by file generateimage.php   You may need to edit php.ini and elsewhere to allow script to read the subdirectory ttf or modify/simplify the code so it does not have to get a directory listing to determine what ttf fonts it can choose. Also note that we set the output of generateimage.php to be jpeg but you may could use png. Note that the imagegettftext [12] function allows us to rotate the text!

Q. How about ASCII art instead of an image?

A.   Added that option with version 0.2 and play with settings in the config.php file

Q. Can I make it harder to OCR by changing the background?

A. Look at generateimage.php and you will see a conditional that if $BASaddimgnoise = "yes" then the script will create random arcs over the background in the font color and an off color. To engage, edit config.php and change the variable $BASaddimgnoise to be "yes". We think that something better could be done and welcome contributions, but if possible we request that the noise method work with versions of GD prior to 2.x so that limits some of the php image commands that may be more convenient for drawing. For example, the function imagefilledellipse [13] was added in PHP 4.0.6 and requires GD 2.0.1 or later

Q. The characters are to hard to read, and I sometimes can't tell difference between number one and some letters, what can I do?

A. You could modify the script to use numbers only on the random generation. Also you could reduce the ttf fonts to choose from, and or replace with ttf files that have easier to read fonts. Many sites using image verification are only using numbers that don't randomly rotate the characters but instead have an image with static dots, lines and

curves. In our example we went to the extreme selecting what we thought to be hard to OCR fonts that the license was interpreted as freely distributable for commercial use.

Q. Why did you use the verbose <script language="php"> tags instead of <?

A. We recommend you don't be lazy on the tags if the files may be edited in by other applications. Programs like Dreamweaver and Frontpage know how to handle <script language="xyz">blah </script>, however they can not display in a design or preview view with the abbreviated tagging of PHP.

Q. Why gd?

A. GD is available on most php installations. We think other applications like NetPBM and ImageMagick which are used by Gallery [14] which is php may give more resources and writers may want to consider that.

Q. Why did you write such sloppy code, why not all functions and classes?

A. It was a hack with spare time not elegance, but with lots of comments. Step forward and rewrite it and donate it back or under the BSD license take the code and redo or rather fork the works. Let us know and we will link.

Q. Why not use another figlet class that is more recently supported like the class in PEAR?

A. The included php figlet class is about two years old and the author does not support it on the mentioned website, but it seems to be working. Some php sites are not setup with PEAR, so we used this simpler file. Note the license for the php filget class not by us and if in the future someone complains we will have to reference it or remove it. Perhaps in the future we will go with PEAR or other figlet if need be. See http://www.figlet.org/ [15] for more information about figlet.

Q. How about an audio captcha?

Here is an article of someone using an audio captcha: http://www.ejeliot.com/pages/2 [16]

Source URL: https://cocoavillagepublishing.com/development/tools/php/scripts

Links
[1] https://cocoavillagepublishing.com/system/files/blockautosubmit_v0.2.tar.gz

[2] https://cocoavillagepublishing.com/system/files/blockautosubmit_v0.1.tar.gz

[3] http://www.opensource.org/

[4] http://en.wikipedia.org/wiki/Captcha

[5] http://en.wikipedia.org/wiki/Turing_test

[6] http://www.activistinabox.org/blockautosubmit/

[7] http://www.php.net/

[8] http://www.boutell.com/gd/

[9] http://old.cocoavillagepublishing.com/us/contact/

[10] http://www.freetype.org/

[11] http://www.cocoavillagepublishing.com/development/tools/php/scripts/blockautosubmit/license.txt

[12] http://us3.php.net/manual/en/function.imagettftext.php

[13] http://us3.php.net/manual/en/function.imagefilledellipse.php

[14] http://gallery.menalto.com/

[15] http://www.figlet.org/

[16] http://www.ejeliot.com/pages/2