Captcha

Extension Key: **captcha**

Copyright 2005, Kasper Skårhøj, <kasperYYYY@typo3.com>

This document is published under the Open Content License

available from http://www.opencontent.org/opl.shtml

The content of this document is related to TYPO3

- a GNU/GPL CMS/Framework available from www.typo3.com

Table of Contents

Introduction

What does it do?

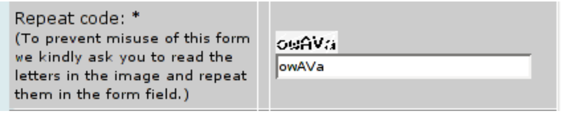
This extension generates an image which contains an obfuscated text string that a user has to repeat in a form field in order to gain access to a website service. The technique assumes that humans can read an repeat the string while a spam-bot cannot, thus preventing guestbooks, tip-a-friend forms etc from being spammed by non-human clients.

The techniques are discussed here: http://en.wikipedia.org/wiki/Captcha

This captcha is meant as a resource other TYPO3 plugins can use. In doing so the overall site security level is improved when this captcha is improved in security. People interested in captcha security is more than welcome to continue development on this initial implementation!

Screenshots

This is how the captcha is used in “tipafriend” extension:



Repeat code: *
(To prevent misuse of this form we kindly ask you to read the letters in the image and repeat them in the form field.)

owAVa

The user submitting the form must repeat the letters in the image in the field below it. If the string is right the form is submitted. If the string is wrong the form is shown again with an error and the user will have another chance to enter the string.

Using the Captcha

In your frontend plugin you simply include an image tag for the captcha image and a form field in addition for entering the code. Then, upon reception of the data you will find the captcha string stored in a session variable and you simply compare that value with whatever is found in the input field. If the match, all is fine and you can process the data.

Showing the image

Very easy:

```
if (t3lib_extMgm::isLoaded('captcha')) {
      $captchaHTMLoutput = '<img src="'.t3lib_extMgm::siteRelPath('captcha').'captcha/captcha.php"
alt="" />';
} else {
      $captchaHTMLoutput = '';
}
```

Showing the response field

Any field name that fits you, for example:

```
<input type="text" size=30 name="TIPFORM[captchaResponse]" value="">
```

Evaluating

First, you need to retrieve the captcha string from the session variable where the captcha.php script generates it:

```
if (t3lib_extMgm::isLoaded('captcha')) {
      session_start();
      $captchaStr = $_SESSION['tx_captcha_string'];
      $_SESSION['tx_captcha_string'] = '';
} else {
      $captchaStr = -1;
}
```

The bold lines are the important ones where the session string is read and subsequently reset so it cannot be used more than once. In this case the captchaString is set to “-1” if the captcha extension is not enabled.

All that is left is checking the captcha string. This can be done with a conditional like this:

```
if (... ($captchaStr===-1 || ($captchaStr && $tipData['captchaResponse']===$captchaStr)) ...) {
...
}
```

Assuming that the input from the form field “TIPFORM[captchaResponse]” is found in “$tipData['captchaResponse']” then the bold part of this condition will check that they match (and is not unset in which case cookies might be disabled).

The acceptance of the captchaString alternatively being “-1” is merely a fallback support in case the “captcha” extension is not installed. So without the captcha extension installed the input will just be accepted. If the captcha extension is installed validation is required. Of course you can altogether require the captcha extension for your extension as a dependency if you do not want people to “run the risk” of not having this security level implemented.