Preventing CSRF

CSRF is an abbreviation for cross-site request forgery, where a malicious user tricks the user’s browser into silently performing an HTTP request to the website when the user is logged in.

An example of such an attack is inserting an invisible image tag with src pointing to http: //example. com/site/logout. Even if the image tag is inserted in another website, you will be immediately logged out from example. com. The consequences of CSRF can be very serious: destroying website data, preventing all website users from logging in, exposing private data, and so on.

Some facts about CSRF:

* As CSRF should be performed by the victim user’s browser, the attacker cannot normally change the HTTP headers sent. However, there are both browser and Flash plugin vulnerabilities that exist which allow users to spoof headers, so we should not rely on these.
* The attacker should pass the same parameters and values as the user would normally.

Considering these, a good method of dealing with CSRF is by passing and checking a unique token during form submissions and, additionally, using GET according to the HTTP specification.

Yii includes built-in token generation and token checking. Additionally, it can automate inserting a token into the HTML forms.

In order to avoid CSRF, you should always:

* Follow, HTTP specification, that is, get should not change its application state
* Keep Yii CSRF protection enabled

In this recipe, we will see how to make sure our application is CSRF-resistant.

Getting ready

Create a new application by using the Composer package manager, as described in the official guide at <http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.

How to do it...

1. In order to turn ON the anti-CSRF protection, we should add config/main. php as follows: 'components' => [ request => [

'enableCsrfValidation => true,

],

],