SYNACKTIV
DIGITAL SECURITY

Cross-Site Scripting in CodeIgniter version 3.1.13

Security advisory

2022-11-29

Antoine Cervoise
Maxime Rinaudo

Vulnerabilities description

CodeIgniter

CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications.[1]

The issues

Synacktiv discovered two reflected Cross-Site Scripting vulnerabilities in error messages handling in:

- Firstly, through error messages generated on failed file includes.
- Secondly, through error messages generated when SQL queries are malformed.

Affected versions

At the time this report is written, the version 3.1.11 and 3,1,13 was proven to be affected.

Timeline

Date	Action
2021-06-25	Advisory sent to security@codeigniter.com but no MX record exists for this domain.
2021-07	Report on the official Slack about issue with the MX record.
2021-10-13	Advisory sent to security@codeigniter.com.
2022-03-11	Agreement between Code Igniter security team and Synacktiv in order to publish a pull request.
2022-03-12	Pull request on the official project (https://github.com/bcit-ci/CodeIgniter/pull/6113).
2022-11-29	Public release

1 https://codeigniter.com/

Technical description and proof-of-concept

1. Reflected Cross-Site Scripting (XSS) in file include error messages

The framework allows loading a file using the following code:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI_Controller {

	public function index()
	{
		$this->load->view('welcome_message');
	}
}
```

If the file to load is based on a user input, an error will be returned if the file does not exist:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI_Controller {

	public function index()
	{
		if (isset($_GET['page'])) {
			$this->load->view($_GET['page']);
		 }
	}
}
```

In this scenario, it is possible for any user to inject HTML and JavaScript into the error message:

```
http://localhost/bcit-ci-CodeIgniter-b73eb19/?page=%3Cscript%3Ealert(%27XSS%20by
%20Synacktiv%27);%3C/script%3E
```

The vulnerability can also be triggered using the dedicated library for fetching GET parameters:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI_Controller {

	public function index()
	{
		$page = $this->input->get('page');
		if (isset($page)) {
			$this->load->view($page);
		 }
	}
}
```

The configuration of the project in *production* mode will not disable this error message.

Recommendation

Sanitize user inputs within error messages using the *xss_clean()* function or apply the patch reported in the following issue: *https://github.com/bcit-ci/CodeIgniter/pull/6113*.

2. Reflected Cross-Site Scripting (XSS) in SQL error messages

In *debug* mode, the framework allows enabling error messages when an SQL query fails in *application/config/database.php*. By default, the *debug* mode is only set on non-production environment:

```
'db_debug' => (ENVIRONMENT !== 'production'),
```

A developper may also force it:

```
'db_debug' => TRUE,
```

With this feature enabled, an attacker able to force an SQL error could inject HTML and JavaScript into the error message. For example, this code sample introduces the vulnerability:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI_Controller {

	public function index()
	{

		$this->load->database();
		if (isset($_GET['id'])) {
			$sql = 'SELECT * FROM toto WHERE id=' . $_GET['id'] . ' LIMIT -1';
			//echo $sql;
			$query = $this->db->query($sql);
			$this->db->error();
		}

		$this->load->view('welcome_message');
	}
}
```

In this scenario, it is possible to inject HTML and JavaScript into the error message:

```
http://localhost/bcit-ci-CodeIgniter-b73eb19/?id=%3Cscript%3Ealert(%27XSS%20by%20Synacktiv
%27);%3C/script%3E
```

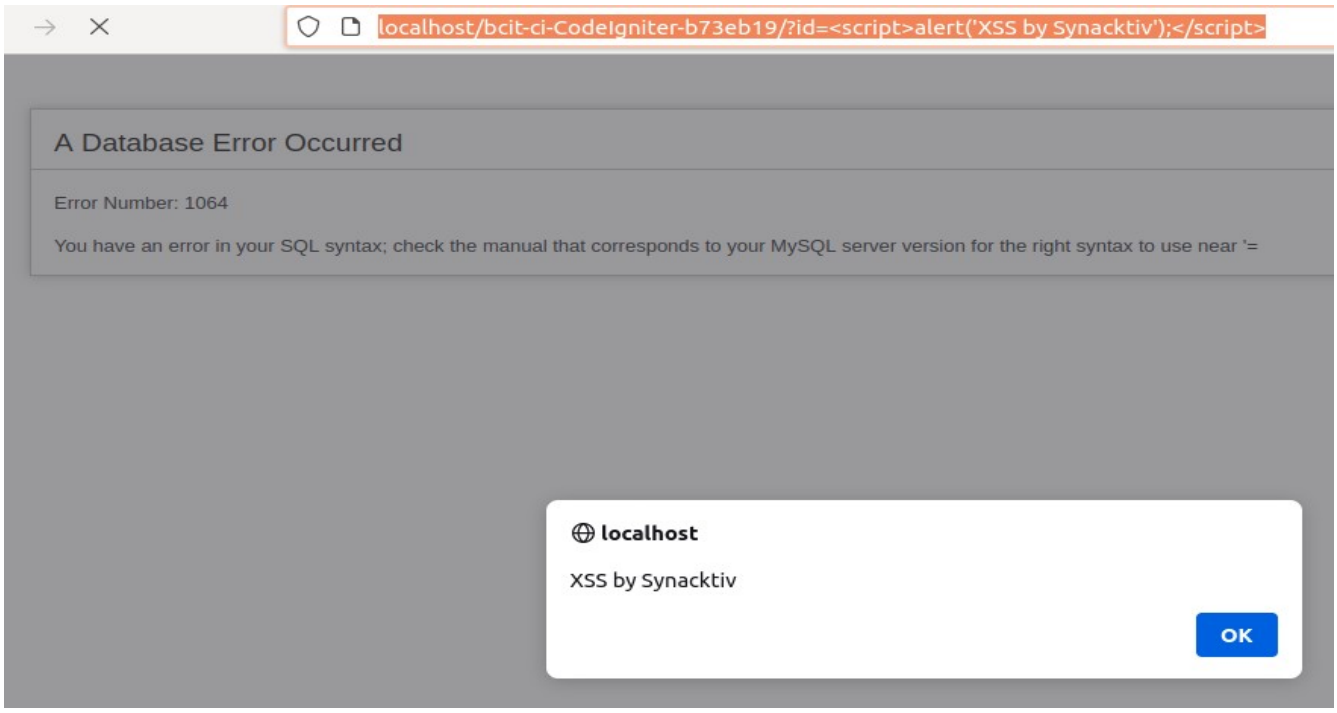

Illustration 1: XSS vulnerability triggered.

Recommendation

Sanitize user inputs within error messages using the *xss_clean()* function or apply the patch reported in the following issue: *https://github.com/bcit-ci/CodeIgniter/pull/6113*.

Configuring the application in *production* mode disables the error message.