Detect SQL Injection WordPress Plugin using regex

Submit by: SunCSR (Sun* Cyber Security Research)

Abstract

A plugin is a piece of software containing a group of functions that can be added to a WordPress website. They can extend functionality or add new features to your WordPress websites. WordPress plugins are written in the PHP programming language and integrate seamlessly with WordPress. In the WordPress community, there is a saying that goes around: “there’s a plugin for that”. They make it easier for users to add features to their website without knowing a single line of code. There are thousands of WordPress plugins available for free at the official WordPress plugin directory. At WPBeginner, we write about all the best WordPress plugins. We have even shared a list of all the plugins we use. Aside from free plugins, there are tons of amazing commercial ones available from third-party companies and developers.

Because it is a software, it still has the risk of security vulnerabilities such as: SQL Injection, Cross Site Scripting (XSS), Cross-Site Request Forgery (CSRF)...

Introduction

Attackers machine: Kali Linux
Victims machine: Windows 10
Tools Used : XAMPP, Burp Suite, SQLmap
Other Requirements : WordPress core and WordPress Plugin

Source code Analysis

SQL Injection occurs when user input is not filtered for escape characters and is then passed into an SQL statement. This results in the potential manipulation of the statements performed on the database by the end-user of the application. The SQL statement is constructed by concatenation before it is passed to function to execute, meaning we are vulnerable to maliciously crafted parameters.

Example of using concatenation in SQL statement:

```
// The user we want to find.
String email = $_REQUEST[ 'email' ]
Connection conn = DriverManager.getConnection(URL, USER,
PASS);
Statement stmt = conn.createStatement();
String sql = "SELECT * FROM users WHERE email = '" + email +
"'";
ResultSet results = stmt.executeQuery(sql);

while (results.next()) {
  // ...oh look, we got hacked.
}
```

In example, email variable is passed to SQL statement by using concatenation: `“email = '" + email + "'";”`

It can raise SQL Injection when email is input from user without any validate.

Now, using regex to find SQL Injection vulnerability in WordPress Plugin Exploit SQL Injection Source code review Using regex to find SQL Query that Plugin Wordpress using to query database:

```
(?<!prepare)\(('|")SELECT.+FROM.+('|").*\..*
```

This regex will find all SELECT query in Plugin without prepare function. Because prepare() function is used to protect queries against SQL Injection attacks.

```
wpdb::prepare( string $query, mixed $args )
```

Bug in real

Using regex `(?<!prepare)\(('|")SELECT.+FROM.+('|").*\..*` to find some bug:

1. Official MailerLite Sign Up Forms < 1.4.4 - Unauthenticated SQL Injection

Affected code:

File: mailerlite-admin.php

```
$form = $wpdb->get_row(
               "SELECT * FROM " . $wpdb->base_prefix
               . "mailerlite_forms WHERE id = " .
$_POST['form_id']
          );
```

Proof of Concept Param "form_id" is vulneable to SQL Injeciton.

`$_POST['form_id']` is directly used in the SQL query, which causes the SQL injection vulnerability

Reference https://wpvulndb.com/vulnerabilities/10235

2. SQL injection in the AdRotate 5.8.3.1 for WordPress exists via param "id"

Affected code:

File: adrotate.php:

```
if(isset($_GET['id'])) $id = esc_attr($_GET['id']);
```

Value of id variable is set by`$_GET['id']`that input by user After that, id variable value is passed to SQL query below:

File: adrotate-statistics.php

```
$stats = $wpdb->get_results("SELECT * FROM
'{$wpdb->prefix}adrotate_stats' WHERE 'ad' = {$id} ORDER BY 'id'
ASC;");
```

Proof of Concept

Parameter" is vulnerable to SQL injection.

Example:

by using a boolean-based technique, one can extract info about the system.

```
http://example.com/wp-admin/admin.php?page=adrotate-statistics&view=group&id=2+AND+1%3D(SELECT+IF+(+GREATEST(+ORD(MID(%40%40version%2C+1%2C+1))%2C+1)+%3D+53%2C+1%2C+0))
```

This query will check if the first char of MySQL version is "5" or not.

More bugs

Some other bug find by SunCSR team by using regex: Blog2Social: Social Media Auto Post & Scheduler < 6.3.1 - Authenticated SQL Injection:

https://wpvulndb.com/vulnerabilities/10260

Form Maker by 10Web < 1.13.36 - Authenticated SQL Injection:

https://wpvulndb.com/vulnerabilities/10237

Photo Gallery by 10Web < 1.5.55 - Unauthenticated SQL Injection:

https://wpvulndb.com/vulnerabilities/10227

Conclusion

If you write SQL query, don’t use the string concatenation. Using parameterized statement for SQL query, parameterized statements make sure that the parameters (i.e. inputs) passed into SQL statements are treated in a safe manner.