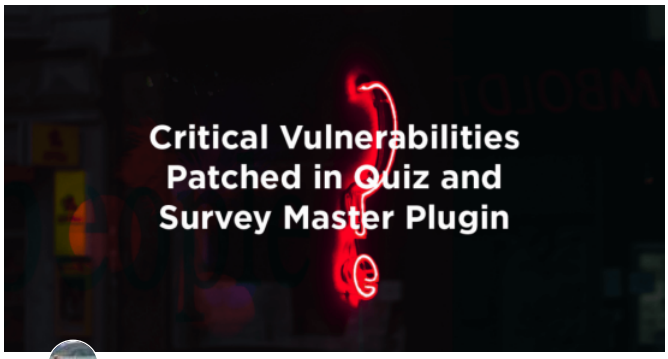

Chloe Chamberland

August 13, 2020

Critical Vulnerabilities Patched in Quiz and Survey Master Plugin

On July 17, 2020, our Threat Intelligence team discovered two vulnerabilities in Quiz and Survey Master (QSM), a WordPress plugin installed on over 30,000 sites. These flaws made it possible for unauthenticated attackers to upload arbitrary files and achieve remote code execution, as well as delete arbitrary files like a site's `wp-config.php` file which could effectively take a site offline and allow an attacker to take over the vulnerable site.

We initially reached out to the plugin's team on July 17, 2020 through their support forum and followed up again on July 21, 2020. After another week of no response, we reached out to ExpressTech, the plugin's parent company, on July 28, 2020, via the contact form on their site providing an end-of-week deadline for response prior to escalating our disclosure. They responded on August 1, 2020 confirming the correct disclosure inbox, and we sent the full disclosure details over on Monday, August 3, 2020. A patch was released just a few days later on August 5, 2020.

We highly recommend updating to version 7.0.1 immediately to keep your site protected against any attacks attempting to exploit this vulnerability.

Wordfence Premium users, and sites still using the free version of Wordfence, have been protected against most exploits targeting these vulnerabilities due to the Wordfence firewall's built-in rules protecting against malicious file uploads, local file inclusion, and directory traversal.

While testing the patch released by the QSM team on August 5, 2020, we discovered our firewall did not provide adequate protection against this vulnerability in the event of a Cross-Site Request Forgery (CSRF) exploit attempt targeting administrators. Though it is less likely that these vulnerabilities would be exploited this way, we quickly created a rule to provide optimal protection. Premium users received this new firewall rule on August 5, 2020. Sites still using the free Wordfence plugin will receive this rule on September 5, 2020.

Description: Arbitrary File Upload
Affected Plugin: Quiz and Survey Master
Plugin Slug: quiz-master-next
Affected Versions: <= 7.0.0
CVE ID: CVE-2020-35949
CVSS Score: 10.00 (CRITICAL)
CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Fully Patched Version: 7.0.1

Quiz and Survey Master is an easy to use plugin designed to add quizzes and surveys to a WordPress site. One feature allows site owners to implement file uploads as a response type for a quiz or survey, which could be useful in a number of scenarios, such as a job application questionnaire with a PDF resume upload at the end.

Unfortunately, this feature was insecurely implemented. The checks performed prior to allowing a file to upload only evaluated how the general settings were configured for the file upload question. For example, it checked what file-type was selected to be "allowed" and the file size permitted, per user specified settings.

```
65  public function qsm_upload_image_fd_question(){
66      global $mlwQuizMasterNext;
67      $question_id = isset($_POST['question_id']) ? sanitize_text_field($_POST['question_id']) : 0;
68      $file_upload_type = $mlwQuizMasterNext->pluginHelper->get_question_setting($question_id, 'file_upload_type');
69      $file_upload_limit = $mlwQuizMasterNext->pluginHelper->get_question_setting($question_id, 'file_upload_limit');
70      $mimes = array();
71      if($file_upload_type){
72          $file_type_exp = explode(',', $file_upload_type);
73          foreach ($file_type_exp as $value) {
74              if($value == 'image'){
75                  $mimes[] = 'image/jpeg';
76                  $mimes[] = 'image/png';
77                  $mimes[] = 'image/x-icon';
78                  $mimes[] = 'image/gif';
79              }else if($value == 'doc'){
80                  $mimes[] = 'application/msword';
81                  $mimes[] = 'application/vnd.openxmlformats-officedocument.wordprocessingml.document';
82              }else if($value == 'excel'){
83                  $mimes[] = 'application/excel, application/vnd.ms-excel, application/x-excel, application/x-msexcel';
84                  $mimes[] = 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet';
85              }else{
86                  $mimes[] = $value;
87              }
88          }
89      }
```

The check to verify file type only looked at the "Content-Type" field during an upload, which could be easily spoofed. This meant that if a quiz contained a file upload which was configured to only accept .txt files, an executable PHP file could be uploaded by setting the "Content-Type" field to 'text/plain' to bypass the plugin's weak checks.

Ultimately, this meant that unauthenticated users could upload arbitrary files, including PHP files, to a site and achieve remote code execution when there was a quiz enabled on the site that allowed file uploads as a response. This could lead to complete site takeover and hosting account compromise amongst many other scenarios.

Fortunately, the functionality had to be enabled and configured for a quiz in order to be exploitable, meaning that most sites were unlikely to be exploited by this particular vulnerability. However, sites with contributor-level, or higher, users with access to the quiz maker tools could still be vulnerable to exploits targeting this vulnerability if an account were compromised. A compromised user account could allow an attacker to create quizzes allowing file uploads achieving the same goal.

Remember to only provide trusted users with access levels greater than subscriber-level access and to enforce strong passwords on these roles so that attackers can't use these accounts as a means of intrusion.

Description: Unauthenticated Arbitrary File Deletion
Affected Plugin: Quiz and Survey Master
Plugin Slug: quiz-master-next
Affected Versions: <= 7.0.0
CVE ID: CVE-2020-35951
CVSS Score: 9.9 (CRITICAL)

CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:H
Fully Patched Version: 7.0.1

were uploaded during the quiz. This `qsm_remove_file_fd_question` function is registered with a regular AJAX action and a `nopriv` AJAX action. This meant that the function could be triggered by unauthenticated users, which is to be expected due to the quizzes not requiring authentication.

```
50 | add_action('wp_ajax_qsm_remove_file_fd_question', array($this, 'qsm_remove_file_fd_question'));
51 | add_action('wp_ajax_nopriv_qsm_remove_file_fd_question', array($this, 'qsm_remove_file_fd_question'));
```

Unfortunately, there were no checks when verifying that the `file_url` supplied for file deletion was from a quiz or survey upload, so any file could be supplied and subsequently removed.

```
137 | public function qsm_remove_file_fd_question(){
138 |     $file_url = isset($_POST['file_url']) ? sanitize_text_field($_POST['file_url']) : '';
139 |     if($file_url){
140 |         wp_delete_file($file_url);
141 |         $json['type']= 'success';
142 |         $json['message'] = 'File removed successfully';
143 |         echo json_encode($json);
144 |         exit;
145 |     }
146 |     $json['type']= 'error';
147 |     $json['message'] = 'File not removed';
148 |     echo json_encode($json);
149 |     exit;
150 | }
```

This made it possible for attackers to delete important files like a site's `wp-config.php` file. Deleting the `wp-config.php` file would disable a site's database connection and allow an attacker to re-complete the installation procedures to connect their own database to a site's file system and regenerate a `wp-config.php` file. At that point they could use this access to infect other sites on the site's hosting account, or continue to use the site to infect site visitors.

Due to this being an unauthenticated AJAX action, the file deletion could be triggered by anyone at any point, regardless of whether or not a site had a quiz accepting uploads enabled.

Proof of Concept Walkthrough

Disclosure Timeline

July 17, 2020 – Initial discovery of vulnerabilities. We determine the Wordfence firewall's built-in rules provide sufficient protection against unauthenticated attackers attempting to exploit these vulnerabilities. Initial attempt to reach out to the QSM plugin team.
July 21, 2020 – Follow-up as previous message deleted in forum.
July 28, 2020 – Attempt to reach out via another contact method found on parent company ExpressTech.io site.
August 1, 2020 – Plugin developer confirms appropriate inbox for handling discussion.
August 3, 2020 – We provide full disclosure details.
August 5, 2020 – A patch is released in version 7.0.1. While reviewing the patch, we discover an additional method an attacker could have used to exploit the vulnerability that our firewall did not cover. We immediately developed a firewall rule and released it to Wordfence Premium users.
September 5, 2020 – Wordfence free users receive the new firewall rule.

Conclusion

In today's post, we detailed two flaws in the Quiz and Survey Master plugin that provided unauthenticated users with the ability to upload arbitrary files and delete arbitrary files. These flaws have been fully patched in version 7.0.1. We recommend that users immediately update to the latest version available, which is also version 7.0.1 at the time of this publication.

Sites using Wordfence Premium and those still using the free version of Wordfence are protected from most attacks against this vulnerability. Wordfence Premium users have been protected against CSRF attacks since August 5, 2020. Sites still running the free version of Wordfence will receive the additional firewall rule protecting against CSRF exploit attempts on September 5, 2020.

If you know a friend or colleague who is using this plugin on their site, we highly recommend forwarding this advisory to them to help keep their sites protected as this is a critical security update.
Did you enjoy this post? Share it!

Comments

1 Comment

Kadigan *
August 14, 2020
2:35 am

Sadly, handling file uploads has always been an issue - short of actually attempting to parse the file's contents for validity (and sometimes not even then, see below), it's often impossible to be 100% sure as to what sort of file was actually uploaded.

Many a code examples often naively rely on built-in functionalities (like MIME type verification provided by hosting web server code) - functionalities that have known limitations and have been known to be unreliable for decades. And even if a file were to parse successfully, it may still be dangerous -- as would be the case with text-padded PHP files (it would pass text/plain verifications easily).

I think that, in the end, there's only one real solution here: always be sure where the uploads are being placed (incl. directory traversal protection), and always have execution capabilities disabled on the upload location. This wouldn't protect against client-side attacks (like uploading JS files) and wouldn't prevent your site being used simply as remote-storage for attacks on other sites, but it should go a long way towards protecting your site from a takeover through malicious uploads.

Breaking WordPress Security Research in your inbox as it happens.

you@example.com

☐ By checking this box I agree to the terms of service and privacy policy.*

Our business hours are 9am-8pm ET, 6am-5pm PT and 2pm-1am UTC/GMT excluding weekends and holidays.
Response customers receive 24-hour support, 365 days a year, with a 1-hour response time.

Terms of Service Privacy Policy

CCPA Privacy Notice

Products

Wordfence Free
Wordfence Premium
Wordfence Care
Wordfence Response
Wordfence Central

Support

Documentation
Learning Center
Free Support
Premium Support

News

Blog
In The News
Vulnerability Advisories

About

About Wordfence
Careers
Contact
Security
CVE Request Form

Stay Updated

Sign up for news and updates from our panel of experienced security professionals.

you@example.com

☐ By checking this box I agree to the terms of service and privacy policy.*

SIGN UP

© 2012-2022 Defiant Inc. All Rights Reserved