Chloe Chamberland

January 8, 2020

Multiple Vulnerabilities Patched in Minimal Coming Soon & Maintenance Mode – Coming Soon Page Plugin

A few weeks ago, our threat intelligence team discovered several vulnerabilities present in Minimal Coming Soon & Maintenance Mode – Coming Soon Page, a WordPress plugin installed on over 80,000 websites. The most severe weakness allowed for an attacker to exploit Cross Site Request Forgery (CSRF) and enable maintenance mode while injecting cross-site scripting (XSS), in addition to several important settings modifications. We later found additional weaknesses that allowed any authenticated user to enable/disable maintenance mode, export settings, and change maintenance mode themes.

We privately disclosed the issue to the plugin's developer, with whom we were already working on a security issue in 301 Redirects – Easy Redirects Manager. As we saw with 301 Redirects, they were quick to acknowledge the report and start working on a patch.

For the vulnerabilities present in Minimal Coming Soon & Maintenance Mode, Wordfence Premium customers received new firewall rules to protect against exploits; free users will receive these rules after thirty days, on February 2, 2020.

Description: CSRF to Stored XSS and Setting Changes
Affected Plugin: Minimal Coming Soon & Maintenance Mode – Coming Soon Page
Affected Versions: <= 2.10
CVE ID: CVE-2020-6167
CVSS Score: 9.6 (Critical)
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H
Patched Version: 2.15

Nonce Checks Needed to Validate Setting Changes

The Minimal Coming Soon & Maintenance Mode – Coming Soon Page plugin provides a plethora of features to help customize a site's maintenance or coming soon page, all of which are accessible in a centralized settings area.

Unfortunately, this plugin had no nonce checks on any of the settings to verify that a request came from a legitimate source, such as a logged in administrative user. The lack of nonce checks created a CSRF vulnerability, and an attacker could craft a request disguised by a link to trick a site owner into modifying the settings of the plugin.

Vulnerable code snippet:

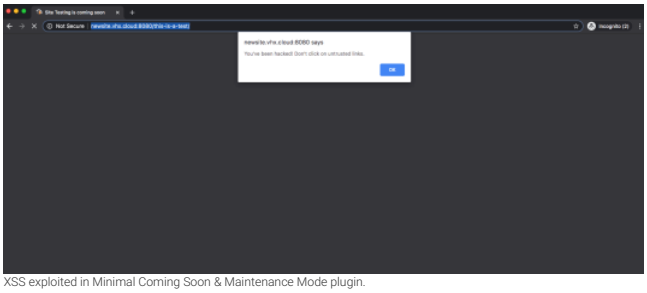
```
51 | } elseif ( isset( $_POST['signals_csmm_submit'] ) ) { {
```

Revised code snippet with nonce verification:

```
51 | } elseif ( isset( $_POST['signals_csmm_submit'] ) && isset($_POST['csmm_save_nonce']) && wp_verify_nonce($_POST['csmm_
```

The functionality of the settings significantly increases the severity of this vulnerability. In this instance, every setting controlling the plugin's features could be modified. This included features like inserting custom HTML, enabling maintenance mode, IP whitelisting, general content design, and importing logos.

An attacker capable of tricking an administrator into clicking on a link with a specially crafted request could create havoc for site owners and their visitors. A malicious link could take the vulnerable site offline by enabling maintenance mode while injecting a malicious javascript into the custom HTML field. That malicious script would then execute when an innocent user browsed the site. This XSS vulnerability could redirect site visitors to malicious websites, infect vulnerable computers, or perform other malicious actions.



XSS exploited in Minimal Coming Soon & Maintenance Mode plugin.

An attacker could also make several additional impactful changes like enabling the "Temporarily Pause Search Engines" setting, hurting a site's search engine ranking, or including remote files as a "logo" on the site, with little to no restriction on file type.

This vulnerability is similar to what we saw in another maintenance mode plugin, WP Maintenance, a few weeks ago. Though CSRF is hard to protect against, the Wordfence firewall's built-in XSS protection protects against any XSS attempts made during a CSRF exploit attempt. Avoid CSRF attacks by not clicking on links or attachments from untrusted sources.

Description: Insecure Permissions: Enable and Disable Maintenance Mode
Affected Plugin: Minimal Coming Soon & Maintenance Mode – Coming Soon Page
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:H
Patched Version: 2.15

The Minimal Coming Soon & Maintenance Mode plugin provides users with the capability to enable and disable maintenance mode from the admin bar to make it more convenient for administrators to toggle between the two modes when performing maintenance on a site.

In order to provide this feature, the plugin registers an admin action with an `is_admin()` check right before the registered action. Unfortunately, it is a common misconception that the `is_admin()` function verifies that a request is coming from an administrator logged into the admin dashboard. However, `is_admin()` only checks that the request is being sent to an administrative page. Administrative pages are accessible to any logged in user, not just administrators.

This created a flaw that allowed any authenticated user with subscriber permissions or above the ability to enable and disable maintenance mode on a vulnerable site by sending a simple request. If an attacker was unable to create a subscriber account on a vulnerable website without open registrations, they could attempt to exploit this using CSRF due to the lack of nonce checks.

Vulnerable code snippet:

```
6  class CSMM {
7   static function init() {
8     if (is_admin()) {
9       add_action('admin_action_csmm_change_status', array(__CLASS__, 'change_status'));
10    }
```

```
24 <?php
25        if ($signals_csmm_options['status']== '1') {
26          $action_url = add_query_arg(array('action' => 'csmm_change_status', 'new_status' => 'disabled', 'redirect' =>
27        } else {
28          $action_url = add_query_arg(array('action' => 'csmm_change_status', 'new_status' => 'enabled', 'redirect' => u
29        }
```

Proof of Concept

In order to exploit this vulnerability, an attacker would login as a user with subscriber or above permissions and send the following request to enable maintenance mode:

```
/wp-admin/admin.php?action=csmm_change_status&new_status=enabled&redirect=/wp-admin/
```

Alternatively, a malicious actor could send the following request to disable maintenance mode:

```
/wp-admin/admin.php?action=csmm_change_status&new_status=disabled&redirect=/wp-admin/
```

Wordfence Premium customers have already received new firewall rules to protect against these exploits; free users will receive these rules after thirty days, on February 2, 2020.

Description: Insecure permissions: Export Settings/Theme Change
Affected Plugin: Minimal Coming Soon & Maintenance Mode – Coming Soon Page
Affected Versions: <= 2.15
CVE ID: CVE-2020-6166
CVSS Score: 5.4 (Medium)
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N
Patched Version: 2.17

Another set of features provided by the Minimal Coming Soon & Maintenance Mode plugin includes the ability to export settings and change maintenance mode themes.

As with the previous vulnerability, we see the same error here: `is_admin()` appears to be the improper permission check used to verify that an action is triggered by an administrative user, when it is just checking that the request is being sent to a page within the administrative dashboard.

This created a flaw that would allow any user logged in as a subscriber or above to export the plugin settings as a .txt file or modify the theme of the maintenance page on a vulnerable site.

Example of vulnerable code for settings export:

```
240 function csmm_plugin_admin_init() {
241       if (!is_admin()) {
242         Return;
243 }
```

```
259 add_action('admin_action_csmm_export_settings', 'csmm_export_settings');
```

```
411 function csmm_export_settings() {
412     $filename = str_replace(array('http://', 'https://'), '', home_url());
413     $filename = str_replace(array('/', '\\', '.'), '-', $filename);
414     $filename .= '-' . date('Y-m-d') . '-csmm.txt';
415
416     $options = csmm_get_options();
417     unset($options['none']);
418     $options = apply_filters('csmm_options_pre_export', $options);
419
420     $out = array('type' => 'CSMM', 'version' => csmm_get_plugin_version(), 'data' => $options);
421     $out = json_encode($out);
422
423     header('Content-Type: text/plain');
424     header('Content-Disposition: attachment; filename=' . $filename);
425     header('Expires: 0');
426     header('Cache-Control: must-revalidate');
427     header('Pragma: public');
428     header('Content-Length: ' . strlen($out));
429
430     @ob_end_clean();
431     flush();
432
433     echo $out;
434     exit;
435  } // export_settings
```

Proof of Concept

In order to exploit this vulnerability an attacker would need to login with subscriber or above permissions and send the following request to export the plugin settings:

```
/wp-admin/admin.php?action=csmm_export_settings&redirect=/wp-admin/
```

Alternatively, a malicious actor could send the following request to change the theme:

```
/wp-admin/admin.php?action=csmm_activate_theme&theme=minimal&redirect=/wp-admin/
```

Wordfence Premium customers have already received new firewall rules to protect against these exploits; free users will receive these rules after thirty days, on February 2, 2020.

Timeline

existing XSS rule mitigates the XSS portion of the attack.
December 19th, 2019 – Developer responds and acknowledges issues.
December 25th, 2019 – Developer releases first patch.
January 3rd, 2020 – Discovery of additional security issues disclosed to plugin developer. New firewall rules released for premium Wordfence users.
January 7th, 2020 – Developer acknowledges additional vulnerabilities, begins working on additional fixes.
January 8th, 2020 – Final patch is released.
February 2nd, 2020 – Free users receive firewall rules.

Conclusion

In today's post, we detailed several vulnerabilities present in the Minimal Coming Soon & Maintenance Mode – Coming Soon Page plugin, which included one critical vulnerability. Fortunately, the plugin developer was incredibly quick to respond and release a patch for the vulnerable endpoints. These flaws have all been patched in version 2.17 and we urge users to update to the latest available version as soon as possible.

Sites running Wordfence Premium have been protected from attacks against this vulnerability since January 3, 2020. Sites running the free version of Wordfence will receive the firewall rule update on February 2, 2020 and should update the plugin immediately.

Comments

5 Comments

Hung *
January 8, 2020
5:21 pm

I am wondering if the plugin has been updated to fix the issue, then why wordfence still has to update its firewall rule.

Chloe Chamberland *
January 9, 2020
10:58 am

Hi Hung,

That's a great question! Although we would love for all of the plugin's users to update to the patched version immediately, we know that isn't always possible. By releasing a firewall rule we are protecting our user's that need a little bit of extra time before updating or simply don't know that a critical security update has been made. Essentially, the firewall rule acts as protection for the interim period between when the vulnerability is discovered and when our users get to update.

Hung *
January 9, 2020
7:16 pm

Hi Chloe,

Thanks for your reply.

So overtime, the firewall rule database will become bigger with many old and rarely-activated rules?

For example, 2-3 years from now.

Chloe Chamberland *
January 13, 2020
9:00 am

Hi Hung,

We continuously monitor custom rules and archive them when data concludes that they are no longer actively exploited over an extended period of time. This ensures that the Wordfence WAF remains speedy and provides our users with the most up to date protection.

Marcin *
January 17, 2020
2:12 am

Ah, is_admin() strikes again.

I would campaign to have is_admin() deprecated in favour of admin_page() or a similar name – something that'd make it clear that you're not actually checking permissions.

As a plugin developer - I admit, I did once think is_admin() checked for general admin permissions - maybe not role, but whether the user had the requisite privileges to access the page. It was my fault, obviously, but the name seemed straightforward enough and seemed to do what I needed to happen... so that I didn't immediately verify my assumption (here, the assumption is: "this function acts on the user object", which turns out to be incorrect).

Stay Updated

you@example.com

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

SIGN UP