Back to all articles
UPDATED: 11.24.2021

Multiple Security Vulnerabilities Fixed In Hide My WP by wpWave

Dave
from **patchstack**

Table of Contents

There were multiple security vulnerabilities fixed in the Hide My WP plugin by wpWave which allowed unauthenticated SQL injection and allowed unauthenticated users to retrieve a token to deactivate the plugin.

Do you want to be the first to be alerted about such vulnerabilities? Sign up for Patchstack Community (Free) plan and monitor up to 99 websites for free.

For plugin developers, we have security audit services and Threat Intelligence Feed API for hosting companies.

The premium plugin Hide My WP (versions 6.2.3 and below) suffers from multiple vulnerabilities. The first vulnerability allows any unauthenticated user to perform SQL injection and the second vulnerability also allows any unauthenticated user to retrieve a reset token which can then be used to deactivate the plugin.

The plugin is described as a plugin that helps you hide your WordPress installation from attackers, spammers, and theme detectors. It can also hide your WordPress login URL and allows you to rename your admin URL.

The described issues were fixed in version 6.2.4 after being in contact with the Envato team.

The security vulnerabilities in the Hide My WP plugin

The SQL injection vulnerability in this plugin existed because of how the IP address is retrieved and how it is used inside of a SQL query.

The snippet of code that is vulnerable to SQL injection looks like the following:

```
$user_ip = $this->hmwp_get_user_ip();
$dbips_info = $wpdb->get_var("SELECT `ip` FROM `{$blocked_ips_table}` WHERE `allow`='1
```

The function hmwp_get_user_ip tries to retrieve the IP address from multiple headers, including IP address headers which can be spoofed by the user such as X-Forwarded-For.

An example using CURL would look like the following:

```
curl --location --request GET "https://example.com" --header "X-Forwarded-For: 1' unio
```

The second vulnerability is caused by the following snippet of code:

```
if (isset($_GET['die_message']) && is_admin())
    add_action('admin_init', array(&$this, 'die_message'), 1000);
```

The die_message function, with any irrelevant code removed, looks like the following:

```
    function die_message()
    {
        if (!isset($_GET['die_message']))
            return;

        switch ($_GET['die_message']) {
            case 'new_admin':
                $title = "Custom Admin Path";
                $token = get_option('hmwp_reset_token');
                $reset_url = plugins_url() . '/'. dirname(HMW_FILE) . '/d.php'.'?token
                $content = sprintf(__('<div class="error"><p>Do not click back or clos
                 <br><br><strong> 2) <span style="color: #ee0000">Edit /wp-config.php
                 <p style="color: #ee0000"><strong>If you get locked out of your WordP
                $content .= "<style>input[type='text']{display: block;width: 100%;max-
                $body_email = 'Hello admin,<br/><br/><p style="color:green"><strong>If
                $subject_email = sprintf(__('[%s] Your New WP Login!', self::slug), se
                wp_mail(get_option('admin_email'), $subject_email, $body_email, array(
                break;
            }
        wp_die('<h3>' . $title . '</h3>' . $content);
    }
```

As you can see, the reset token (hmwp_reset_token) will be directly printed onto the screen which can then be used to deactivate the plugin in the file /wp-content/plugins/hide_my_wp/d.php (located in the root folder of the plugin).

Note that this will only work if a valid token actually exists and the token is not an empty value.

Simply by visiting a URL such as /wp-admin/admin-ajax.php?die_message=new_admin&action=heartbeat we can make it display the reset token on the screen.

The patch in Hide My WP

Since this is a premium plugin, the patch cannot be seen at the WordPress.org SVN repository.

Based on our own research, we can confirm that for the SQL injection vulnerability a patch has been applied that validates the IP address using PHP's filter_var function along with the usage of the $wpdb->prepare function.

For the second vulnerability, a capability check has been added to the die_message function using the current_user_can function that makes sure the current user has the manage_options capability.

Timeline

29-09-2021 - We discovered the vulnerability in Hide My WP and released a virtual patch to all Patchstack paid version customers.
29-09-2021 - We reached out to the developer of the plugin.
05-10-2021 - No reply from the developer of the plugin. We reached out to Envato who replied within 10 minutes.
26-10-2021 - The developer released a new plugin version, 6.2.4, which fixes the issues.
24-11-2021 - Published the article.
24-11-2021 - Added the vulnerability to the Patchstack vulnerability database.

Do you have any vulnerable plugins or themes?

Check for free

Get latest WordPress security insight from our **Patchstack Weekly** series

Start listening

Related Articles

View All >

WORDPRESS SECURITY VULNERABILITIES

Most Common WordPress Plugin Vulnerabilities & How to Fix Them

LAST PATCH, WORDPRESS PLUGIN SECURITY

Patching an Arbitrary User Creation Security Bug in "thecartpress" Plugin

PATCHSTACK WEEKLY

Patchstack Weekly #51: How One Vulnerability Affects Many

WordPress security

Plugin auditing

Vulnerability database

Vulnerability API

Bug bounty program

Patchstack for WordPress

For agencies

For hosts

For plugins NEW

Pricing & features

Documentation

Patchstack

About us

Careers

Media kit

Articles & insight

Whitepaper 2021

Social

LinkedIn

Facebook

Twitter

hackuu

Join Discord

DPA Privacy Policy Terms & Conditions © 2022