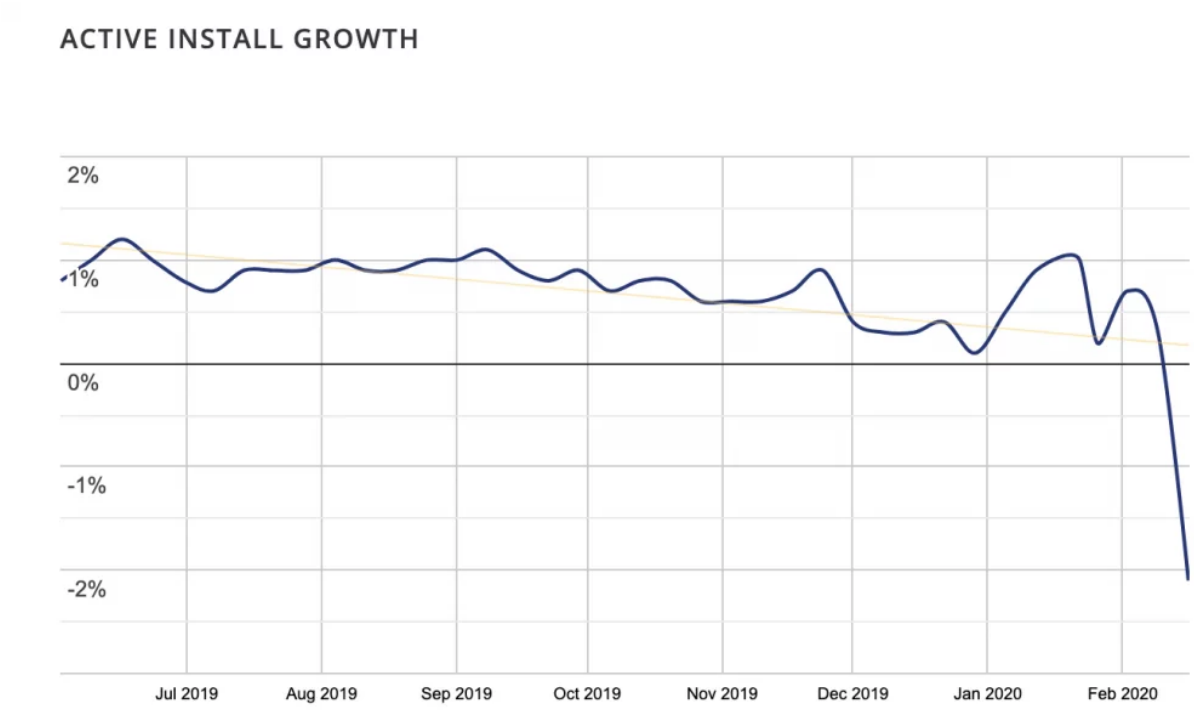
Back to all articles
UPDATED: 02.16.2020

Critical Issue In ThemeGrill Demo Importer

Dave
from **patchstack**

The ThemeGrill Demo Importer plugin has 200,000+ active installations and can be used to import ThemeGrill official themes demo content, widgets, and theme settings with just one click.

Update (18th of February):
The installs count has dropped to 100K+. It indicates that many people have started to uninstall the plugin based on the statistics provided by the WordPress plugin repository. Here's the Google Cache showing 200K+ installs on the 15th of February.



In versions 1.3.4 and above and versions 1.6.1 and below, there is a vulnerability that allows any unauthenticated user to wipe the entire database to its default state after which they are automatically logged in as an administrator.

The prerequisite is that there must be a theme installed and activated that was published by ThemeGrill. In order to be automatically logged in as an administrator, there must be a user called "admin" in the database. Regardless of this condition, the database will still be wiped to its default state.

Based on the SVN commit history, this issue has existed in the code for roughly 3 years, since version 1.3.4.

Once the plugin detects that a ThemeGrill theme is installed and activated, it loads the file */includes/class-demo-importer.php* which hooks *reset_wizard_actions* into *admin_init* on line 44.

The admin_init hook runs not only in the admin environment but also on calls to /wp-admin/admin-ajax.php which does not require a user to be authenticated.

The function *reset_wizard_actions* looks a bit like the following (irrelevant code removed):

```
public function reset_wizard_actions() {
    global $wpdb, $current_user;

    if ( ! empty( $_GET['do_reset_wordpress'] ) ) {

        ///

        if ( 'admin' != $current_user->user_login ) {
            $user = get_user_by( 'login', 'admin' );
        }

        if ( empty( $user->user_level ) || $user->user_level < 10 ) {
            $user = $current_user;
        }

        // Drop tables.
        $drop_tables = $wpdb->get_col( sprintf( "SHOW TABLES LIKE '%s%%'", str_rep
        foreach ( $drop_tables as $table ) {
            $wpdb->query( "DROP TABLE IF EXISTS $table" );
        }

        // Installs the site.
        $result = wp_install( $blogname, $user->user_login, $user->user_email, $bl

        // Updates the user password with a old one.
        $wpdb->update(
            $wpdb->users,
            array(
                'user_pass'           => $user->user_pass,
                'user_activation_key' => '',
            ),
            array( 'ID' => $result['user_id'] )
        );

        // Set up the Password change nag.
        $default_password_nag = get_user_option( 'default_password_nag', $result['
        if ( $default_password_nag ) {
            update_user_option( $result['user_id'], 'default_password_nag', false,
        }

        ///

        // Update the cookies.
        wp_clear_auth_cookie();
        wp_set_auth_cookie( $result['user_id'] );

        // Redirect to demo importer page to display reset success notice.
        wp_safe_redirect( admin_url( 'themes.php?page=demo-importer&browse=all&res
        exit();
    }
}
```

Here we see that there is no authentication check and only the *do_reset_wordpress* parameter needs to be present in the URL on any "admin" based page of WordPress, including /wp-admin/admin-ajax.php.

If we are currently not logged in, it will retrieve the "admin" user object from WordPress and then drop all WordPress tables that start with the defined WordPress database prefix.

Once all tables have been dropped, it will populate the database with the default settings and data after which it will set the password of the "admin" user to its previously known password.

However, this does not matter since we are automatically logged in as "admin" near the end of the function. If the "admin" user does not exist in the database then the users' table will remain empty and you will not be automatically logged in as any user.

The patch can be found here which shows that they added a *current_user_can('manage_options')* check to the reset_wizard_actions method.

This is a serious vulnerability and can cause a significant amount of damage. Since it requires no suspicious-looking payload just like our previous finding in InfiniteWP, it is not expected for any firewall to block this by default, and a special rule needs to be created to block this vulnerability.

Timeline

06-02-2020 - Discovery of the issue and released a patch to all Patchstack customers.
06-02-2020 - Reported the issue to the developer of the plugin.
11-02-2020 - Second attempt to reach out to the developer.
14-02-2020 - Received email from the developer, resent the issue to them.
16-02-2020 - The developer published a new version that fixes the issue.

Indicators of compromise

Plugin changelogs are often monitored by the attackers to detect security bug fixes and to compare different versions to see what was fixed. This allows the attackers to act before the users have updated the plugin. This is why updating the plugins as fast as possible is very important.

We have been closely monitoring the ThemeGrill Demo Importer vulnerability and have seen this vulnerability being exploited since the release of the patch.

Patchstack (formerly WebARX) has blocked over 16,000 attacks against this vulnerability since the 16th of February.

List of IP addresses currently exploiting this vulnerability with 100 or more attacks blocked.
149.202.75.164
192.169.159.241
209.251.53.192
107.180.225.158
62.76.187.179
185.45.72.159
31.207.33.10
198.12.156.154
163.44.207.54
142.44.151.107
68.183.204.202
51.68.124.88
188.166.16.17
168.63.19.216
46.105.92.10
103.221.222.179
104.238.99.130
175.139.199.53
68.183.76.157
45.32.104.33
2001:41d0:d:34a4::
46.101.174.128
2607:5300:61:bd9::107
159.65.65.204
84.238.108.177
188.166.176.184
165.227.48.147
2a03:b0c0:2:d0::11f0:6001
2404:f080:1101:316:163:44:207:54
50.63.162.9

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

All solutions

- WordPress security
- Plugin auditing
- Vulnerability database
- Vulnerability API
- Bug bounty program

WordPress security

- 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