Chloe Chamberland

February 25, 2020

Multiple Vulnerabilities Patched in Pricing Table by Supsystic Plugin

On January 17th, our Threat Intelligence Team discovered several vulnerabilities in Pricing Table by Supsystic, a WordPress plugin installed on over 40,000 sites. These flaws allowed an unauthenticated user to execute several AJAX actions due to an insecure permissions weakness. Attackers were also able to inject malicious Javascript due to a Cross-Site Scripting (XSS) vulnerability, access pricing table data, and forge requests on behalf of a site administrator because of a Cross-Site Request Forgery (CSRF) vulnerability.

These vulnerabilities could allow attackers the ability to run malicious Javascript on a visitor's browser that could redirect site visitors to malicious websites, or even steal user cookies to authenticate as an administrator. We privately disclosed these issues to the plugin's author, who released patches a month later.

We highly recommend updating to version 1.8.2 immediately as these security issues are fully patched in that version.

Wordfence premium users received a new firewall rule on January 18th to protect against exploits targeting these vulnerabilities. Free Wordfence users received this rule on February 17th.

Description: Insecure Permissions on AJAX Actions
Affected Plugin: Pricing Table by Supsystic
Affected Versions: <= 1.8.1
CVE ID: CVE-2020-9392
CVSS Score: 7.3 (High)
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
Patched Version: 1.8.2

Pricing Table by Supsystic provides users with the ability to easily add customizable pricing tables to their site. These can be used to display pricing for products or services and compare the differences between each offering. The plugin makes it easy to create new tables, import or modify tables, and export pricing table settings, all of which are powered by AJAX actions. While analyzing the plugin, we discovered that the AJAX actions were registered with a `wp_ajax_nopriv_` hook, allowing any unauthenticated user the ability to successfully send an AJAX request completing an action registered with that hook.

```
258 | add_action('wp_ajax_nopriv_'. $this->_action, array($mod->getController(), $this->_action));
```

Upon further analysis, we found that despite the the actions being registered with a `wp_ajax_nopriv_` hook, there was a separate security permission check on nearly all of the AJAX actions. However, we found that the permissions check was missing for a few select actions. The actions missing a permission check were: `createFromTpl`, a function that creates new pricing tables, `getJSONExportTable`, a function used to export existing tables, and `importJSONTable`, a function used to import a new pricing table or update an old one.

```
257 | public function getPermissions() {
258 |         return array(
259 |             PTS_USERLEVELS => array(
260 |                                 PTS_ADMIN => array('getListForTbl', 'remove', 'removeGroup', 'clear',
261 |                                     'save', 'exportForDb', 'updateLabel', 'changeTpl', 'saveAsCopy', 'getJSONExportTable'
```

**You can see in this code snippet that `getJSONExportTable`, `importJSONTable`, and `createFromTpl` are missing from the getPermissions check at the bottom of the controller.php file. These were added in the latest version of the plugin.

This meant that any unauthenticated user could execute those 3 functions and obtain sensitive information regarding any given pricing table while creating and importing new pricing tables or altering already existing ones.

Information retrieved from getJSONExportTable.

Description: Unauthenticated Stored XSS
Affected Plugin: Pricing Table by Supsystic
Affected Versions: <= 1.8.1
CVE ID: CVE-2020-9393
CVSS Score: 6.1 (Medium)
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:N
Patched Version: 1.8.2

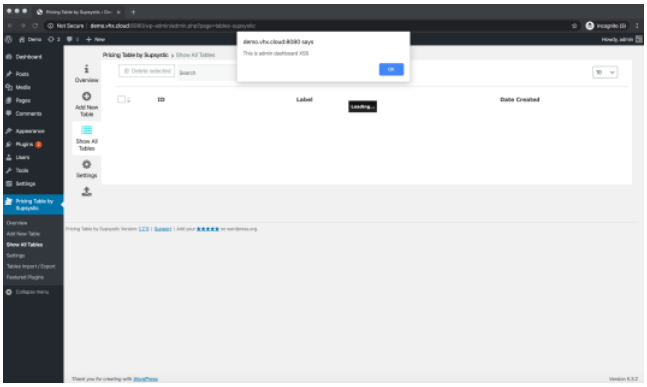
As an extension of the previous vulnerability, we discovered a stored XSS vulnerability could be exploited from the [illegible] endpoint. This function is used to import a JSON body containing all of the settings needed to create

Several of the parameters are handled without any input sanitization, allowing Javascript as an input. Alone, this wouldn't be considered a security issue as user input supplied via the administrative dashboard isn't sanitized due to the fact that administrators have the capability to add `unfiltered_html`. However, when combined with a situation where AJAX actions can be sent with no authentication, an XSS vulnerability is created.

In order to exploit this vulnerability, an attacker would need to send a request containing the details of the table they would like to modify, along with their malicious Javascript payload. This payload could be a script that steals user cookies and sends them off to an attacker for the attacker to gain administrative access to your site. Alternatively, this payload could also be a script that redirects users to a malicious site where their computer will be infected. The malicious javascript would then be executed anytime a user navigated to the page with the stored script.

An attacker could edit a pricing table so that the malicious payload only executed when an administrator accessed the pricing table list from the administrative dashboard, or it could be executed when any user accessed a page that displayed a pricing table. It simply depended on which parameter the malicious Javascript was injected into.

Here is what it would look like if just the name parameter was injected in the data label parameter and executed in the administrative dashboard.



Here is what it would look like if the custom html parameter was injected, executing on the front-end of a site.

Description: Cross-Site Request Forgery to XSS and Setting Changes
Affected Plugin: Pricing Table by Supsystic
Affected Versions: <= 1.8.0
CVE ID: CVE-2020-9394
CVSS Score: 8.8 (High)
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
Patched Version: 1.8.1

To further escalate, we found that none of the endpoints in the plugin were protected by WordPress nonces for CSRF protection. The source of requests were not verified and an attacker could forge a crafted request on behalf of a site administrator and inject malicious Javascript or simply modify the settings of any given pricing table.

PoC Walkthrough: Exploiting XSS

Disclosure Timeline

January 17th, 2020 – Vulnerability initially discovered and analyzed. We begin working on firewall rules.
January 18th, 2020 – Firewall rule released for Wordfence premium users. Initial outreach to plugin team.
January 21st, 2020 – Developer confirms appropriate inbox for handling discussion. Full disclosure of vulnerabilities is sent.
January 30th, 2020 – Follow-up with the developer as no response from disclosure.
February 10th, 2020 – Additional follow-up as no response from disclosure still.
February 11th, 2020 – Developer acknowledges report. Notifies us that the patch will be released the following week.
February 17th, 2020 – Wordfence free users receive firewall rule.
February 21st, 2020 – Patch released. Missing permission check on one action, notified developer.
February 24th, 2020 – Sufficient patch released.

Conclusion

Pricing Table by Supsystic plugin. These flaws have been patched in version 1.8.2 and we recommend that users update to the latest version available immediately. Sites running Wordfence Premium have been protected from attacks against this vulnerability since January 18th. Sites running the free version of Wordfence received the same firewall rule update on February 17th, 2020.

Did you enjoy this post? Share it!

Comments

2 Comments

Charlie *
February 25, 2020
12:48 pm

As always thanks for the great work. We have some customers using this who were all updated to 1.8.2 same day as new version released.

Rajesh Laddha *
March 2, 2020
4:17 am

It is important to secure your WordPress website as it can be vulnerable to hacking. You have mentioned a great technical solution the patched in the pricing table by supsystic plugin. We should keep our website safe from each and every aspect be it plugin or software update.