Chloe Chamberland July 16, 2020

2 Million Users Affected by Vulnerability in All in One SEO Pack

On July 10, 2020, our Threat Intelligence team discovered a vulnerability in All In One SEO Pack, a WordPress plugin installed on over 2 million sites. This flaw allowed authenticated users with contributor level access or above the ability to inject malicious scripts that would be executed if a victim accessed the wp-admin panel's 'all posts' page.

We reached out to the plugin's team the same day of discovery on July 10, 2020 and a patch was released just a few days later on July 15, 2020.

This is considered a medium severity security issue that, as with all XSS vulnerabilities, can result in complete site takeover and other severe consequences. We strongly recommend immediately updating to the latest version of this plugin. At the time of writing, that is version 3.6.2 of All in One SEO Pack.

Wordfence Premium customers received a new firewall rule on July 10, 2020 to protect against exploits targeting this vulnerability. Free Wordfence users will receive this rule after thirty days, on August 9, 2020.

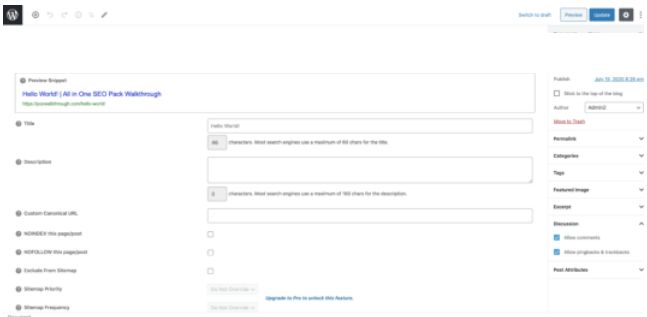
Description: Authenticated Stored Cross-Site Scripting
Affected Plugin: All in One SEO Pack
Plugin Slug: all-in-one-seo-pack
Affected Versions: <= 3.6.1
CVE ID: CVE-2020-35946
CVSS Score: 5.4 (Medium)
CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N
Fully Patched Version: 3.6.2

All In One SEO Pack is a plugin that provides several SEO enhancing features to help rank a WordPress site's content higher on search engines. As part of its functionality, it allows users that have the ability to create or edit posts to set an SEO title and SEO description directly from a post as it is being edited. This makes it easier for post creators to improve the SEO of posts as they are writing them. This feature is available to all users that can create posts, such as contributors, authors, and editors.

Unfortunately, the SEO meta data for posts, including the SEO title and SEO description fields, had no input sanitization allowing lower-level users like contributors and authors the ability to inject HTML and malicious JavaScript into those fields.

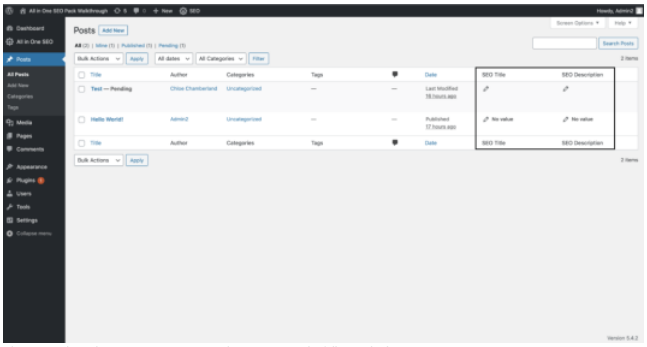
```
5058  /**
5059   * Saves the data of our metabox settings for a post.
5060   *
5061   * @since   ?
5062   * @since   3.4.0   Added support for priority/frequency + minor refactoring.
5063   *
5064   * @param   int     $id     The ID of the post.
5065   * @return  bool            Returns false if there is no POST data.
5066   */
5067  function save_post_data( $id ) {
5068      $awmp_edit = null;
5069      $nonce     = null;
5070
5071      if ( empty( $_POST ) ) {
5072          return false;
5073      }
5074
5075      if ( isset( $_POST['aiosp_edit'] ) ) {
5076          $awmp_edit = $_POST['aiosp_edit'];
5077      }
5078
5079      if ( isset( $_POST['nonce-aioseop-edit'] ) ) {
5080          $nonce = $_POST['nonce-aioseop-edit'];
5081      }
5082
5083      if ( isset( $awmp_edit ) && ! empty( $awmp_edit ) && wp_verify_nonce( $nonce, 'edit-aioseop-nonce' ) ) {
5084
5085          $optlist = array(
5086              'keywords',
5087              'description',
5088              'title',
5089              'custom_link',
5090              'sitemap_exclude',
5091              'disable',
5092              'disable_analytics',
5093              'noindex',
5094              'nofollow',
5095              'sitemap_priority',
5096              'sitemap_frequency',
5097          );
5098
5099          if ( empty( $this->options['aiosp_can'] ) ) {
5100              unset( $optlist['custom_link'] );
5101          }
5102
5103          if ( ! AIOSEOPPRO ) {
5104              $optlist = array_diff( $optlist, array( 'sitemap_priority', 'sitemap_frequency' ) );
5105          }
5106
5107          foreach ( $optlist as $optionName ) {
5108              $value = isset( $_POST[ "aiosp_$optionName" ] ) ? $_POST[ "aiosp_$optionName" ] : '';
5109              update_post_meta( $id, "_aioseop_$optionName", $value );
5110          }
5111      }
5112  }
```

Here is a look at where these fields can be edited in the post editor:



SEO area in post from All in One SEO Pack.

The SEO title and SEO description for each post are always displayed on the 'all posts' page as they appear in the far right column for easier quick editing access. Therefore, any values added to the SEO title and SEO description fields would be displayed here in an unsanitized format, causing saved JavaScript in these fields to be executed when any user accessed the 'all posts' page.



The SEO Title and SEO Description areas that appear on the 'all posts' admin area.

Any JavaScript injected in the SEO description field would also be executed when visiting the page directly if a closing tag was inserted by an attacker before adding their own script. For example, it could look like `</script><script>alert(0)</script>`. This was due to the fact that the tag would close out the SEO description's original script tag and inject an additional script directly after.

Using a closing script tag to start a new script.

Due to the JavaScript being executed whenever a user accessed the 'all posts' page, this vulnerability would be a prime target for attackers that are able to gain access to an account that allows them to post content. Since Contributors must submit all posts for review by an Administrator or Editor, a malicious Contributor could be confident that a higher privileged user would access the 'all posts' area to review any pending posts. If the malicious JavaScript was executed in an Administrator's browser, it could be used to inject backdoors or add new administrative users and take over a site.

Fortunately, in the patched version, the plugin developer has added sanitization to all of the SEO post meta values so any HTML characters supplied will be escaped and unable to become executable scripts.

Proof of Concept Walkthrough

How Concerning Is a Contributor+ vulnerability?

The great news about this vulnerability is that it requires a high level of permissions to exploit, making it more difficult for attackers to actually utilize in an attack. Therefore, it is less likely to be targeted as part of a mass automated campaign. It could, however, be one additional method of escalating a more sophisticated attack. As such, there are a few security precautions you should always remember to take to help protect your site against vulnerabilities targeting higher-level user exploits.

Always use the principle of least privilege.
Least privilege is a security concept that suggests that users be provided with the minimal amount of privileges required to do their job. This means that when you are supplying users with access to your site, you should make sure you are providing them with the least amount of privileges needed to perform any needed actions. If your users don't need to write posts but need an account, make sure you are only providing them with subscriber-level access or equivalent. You can read more about WordPress Roles and Capabilities here.

We understand that in some cases you may need to provide users with slightly more privileges to do certain tasks than their normal day-to-day routine. In these cases, we recommend providing users with temporarily elevated privileges and then revoking those privileges once the task has been completed.

For example, if you want to have someone write a guest post on your WordPress blog, we recommend providing them with contributor-level access to write the post and then, once they are finished, downgrade their privileges to subscriber-level. Once they are done writing the guest post they no longer need contributor-level privileges, therefore, maintaining the principle of least privilege by downgrading those privileges is optimal for your site's security.

Along with this, we recommend auditing your site's user accounts to make sure there are no rogue or left-over accounts that should be deleted. Leaving unused accounts on a site provides attackers with more possible intrusion vectors for exploiting vulnerabilities that require higher-level permissions.

Trust and verification are important when providing users with access to your site.
When providing users with higher-level roles like contributor, author, and editor, we highly recommend verifying that the user can be trusted. This can be done by checking personal references or establishing security protocols limiting access to people who work for reputable companies.

If someone calls you, sends you an email, or contacts you in any way saying they need access to your site for any reason, and it's not coming from someone you know and trust, then it might be a social engineering attempt. Never

Always remember to never share accounts or passwords. Instead, establish separate accounts and choose the option to notify the user about the account via email. In this way, the user can set up their own password, and you can enforce strong passwords using Wordfence, so that passwords are never transmitted via email which should be considered an insecure channel of communication. As you are not sharing accounts, you can always revoke access immediately for a new user if malicious actions start to appear.

Use and enforce strong passwords for end users, especially those with higher privileges.
We highly recommend enforcing strong passwords for all users, however, accounts with higher privileges have an elevated risk as they have more capabilities associated with their account. For that reason, strong passwords are extremely important to enforce, in order to mitigate the risk associated with attackers gaining unauthorized access to these accounts through password compromising attack techniques like brute force.

We also recommend enforcing two-factor authentication for all users, especially those with higher level capabilities, to help provide an extra layer of login security and protection against brute force attacks and compromised passwords. Wordfence makes this easy with built in functionality that can be found in the "Login Security" area of the Wordfence plugin or with the use of the stand alone Wordfence Login Security Plugin. You can learn more on how to enable and configure these settings here.

Disclosure Timeline

July 10, 2020 – Initial discovery and analysis of vulnerability. Firewall rule was released for Wordfence Premium customers. Initial outreach to the Semper plugin team.
July 13, 2020 – The lead developer at Semper confirms an appropriate discussion channel. We provide full disclosure.
July 15, 2020 – A patch was released (version 3.6.2).
August 9, 2020 – Free Wordfence users receive firewall rule.

Conclusion

In today's post, we detailed a flaw that allowed higher-level WordPress users the ability to inject malicious scripts into posts in the All in One SEO plugin. This flaw has been fully patched in version 3.6.2. We recommend that users immediately update to the latest version available at the time of reading this.

Sites running Wordfence Premium have been protected from attacks against this vulnerability since July 10, 2020. Sites running the free version of Wordfence will receive this firewall rule update on August 9, 2020. If you know a friend or colleague using this plugin on their site, we strongly recommend forwarding this advisory to them so that they can update and protect their WordPress site.

Special thank you to the Lead Developer for All in One SEO Pack, Benjamin Rojas, for working quickly to get a patch out to protect users.
Did you enjoy this post? Share it!

Comments

9 Comments

Kyle *
July 16, 2020
10:38 am

Wow! That's really scary to think of the opportunity that a hacker could have based on the properties of those fields in the "All Posts" page in the admin area. Glad you guys are vigilant and looking out for us site owners. Mad respect!

Greg Cogar *
July 16, 2020
2:44 pm

Wordfence alerted me yesterday after a scan run in an auto emails about this update,

* The Plugin "All In One SEO Pack" needs an upgrade (3.6.1 -> 3.6.2).

I always update my websites when Wordfence tells me and glad, thank you very much.

Nisha Garg *
July 17, 2020
12:37 am

Oh weird! That was actually interesting how hackers are trying to inject the script. I did receive email from wordfence to update the plugin but I thought it was a regular update and can be updated later. But, today when I saw the "vulnerability" email from them, I was scared. Anything could have happened. Now, updated all the sites. Thanks.

Pietro *
July 17, 2020
12:53 am

Wow, Thanks for your make us safe everyday with those helpfull discoveries!

Adam Lavery *
July 17, 2020
3:19 am

This is a classic case of why Wordpress needs to enforce new rules on plugin developers, to differentiate security updates from the all too frequent BAU updates that occur. Barely a day goes by without one plugin or another needing an update. There should be a limit on how frequently plugin developers are allowed to release BAU updates.

Issuing separate security-only updates would allow for automatically applying these, as happens with Wordpress security updates.

David R *
July 19, 2020
2:49 pm

Great catch and even better time to response/remediation. The fact contributor+ access is required for exploitation definitely lowers the severity and is a hurdle for potential mass exploitation, but it's great to see the vigilance here. If only more developers were that on top of things threat actors would have far less of a foothold for mass malware campaigns.

Ahmad Shahbaz *
July 20, 2020
4:15 am

Thanks Wordfence for save my sites. I Updated all my sites.

Jay *
July 20, 2020
7:58 am

Thank you Wordfence! I checked two of our sites yesterday. Both were upgraded to v3.6.2 on 7/18/20, and we did NOT do perform these updates. Plugin updates are set to manual on both sites, but Wordpress core security updates are set to automatic.

Is it possible that Wordpress pushed updates for this plugin?

The sites both have Wordfence Premium installed, so we should have been protected from a malicious person spoofing updates. Thank you!

Chloe Chamberland *
July 20, 2020
7:22 pm

Hi Jay!

In some rare cases the WordPress team may force a plugin update due to a security issue if they deem it is significant enough, and if the update can happen safely for all sites. You can read about that here. It sounds like they may have pushed an auto-update for this plugin and that is why your sites updated without any interaction on your part.

Hope that helps!
Chloe