XSS Vulnerability in Social Auto-Poster Plugin Impacts 100,000 Sites
Wordfence

Ram Gall

October 28, 2021

XSS Vulnerability in NextScripts: Social Networks Auto-Poster Plugin Impacts 100,000 Sites

Note: To receive disclosures like this in your inbox the moment they're published, you can subscribe to our[WordPress Security Mailing List](#).

On August 19, 2021, the Wordfence Threat Intelligence team began the disclosure process for a reflected Cross-Site Scripting(XSS) vulnerability we found in NextScripts: Social Networks Auto-Poster, a WordPress plugin with over 100,000 installations.

The plugin's developer responded, so we confidentially provided the full disclosure the next day, on August 20, 2021. After several weeks without updates, we followed up with the developer on September 27, 2021, and a patched version of the plugin, 4.3.21, was released on October 4, 2021.

All Wordfence users, including Wordfence Premium customers as well as those still using the free version of Wordfence, are protected against this vulnerability by our firewall's built-in cross-site scripting protection.

Description: Reflected Cross-Site Scripting(XSS)
Affected Plugin: NextScripts: Social Networks Auto-Poster
Plugin Slug: social-networks-auto-poster-facebook-twitter-g
Affected Versions: <= 4.3.20
CVE ID: CVE-2021-38356
CVSS Score: 6.1(Medium)
CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Researcher/s: Ramuel Gall
Fully Patched Version: 4.3.21

The `nxs_ReposterListTable::column_title` function in `inc/nxs_class_snap.php` echoed out the value of `$_REQUEST['page']` when an administrator was visiting the plugin administration page at `wp-admin/admin.php?admin.php?page=nxssnap-post`.

```
function column_post_title($item){
    //Build row actions
    $actions = array(
        'edit'      => sprintf('<a href="?page=%s&action=%s&item=%s">Edit</a>',$_REQUEST['page'],'edit',$item->ID),
        'delete'    => sprintf('<a href="?page=%s&action=%s&item=%s">Delete</a>',$_REQUEST['page'],'delete',$item->I
    );
    //Return the title contents
    return sprintf('%1$s <span style="color:silver">(id:%2$s)</span>%3$s',
        /*$1%s*/ $item->post_title,
        /*$2%s*/ $item->ID,
        /*$3%s*/ $this->row_actions($actions)
    );
}
```

WordPress uses the value of the `$_GET['page']` parameter in order to determine which administrative page to serve content for. It is also common practice for developers to use `$_REQUEST` for values stored in either `$_GET` or `$_POST`, as the `$_REQUEST` superglobal contains everything set in both `$_GET` and `$_POST`. As such, `$_REQUEST['page']` might be expected to be set to the same value as `$_GET['page']`.

However, thanks to a quirk of how PHP orders parameters that are present in multiple superglobal variables, it was possible to perform a reflected cross-site scripting attack.

In most PHP configurations, depending on the `request_order` (or the `variables_order` if `request_order` is not set), `$_POST` takes precedence over `$_GET` when populating `$_REQUEST`. In other words, if both `$_GET['page']` and `$_POST['page']` are set, `$_REQUEST['page']` will be set to the contents of `$_POST['page']`, rather than `$_GET['page']`.

This meant that it was possible to execute JavaScript in the browser of a logged-in administrator by tricking them into visiting a self-submitting form that sent a `POST` request to their site, for example, `hxxps://victimsite.site/wp-admin/admin.php?page=nxssnap-post`, with the `$_POST['page']` parameter set to malicious JavaScript.

The `$_GET['page']` parameter could be set to `nxssnap-post`, so that WordPress would route the victim to the correct page, and then the malicious JavaScript in `$_POST['page']` would be echoed out on that page.

As with all XSS attacks, malicious JavaScript running in an administrator's session could be used to add malicious administrative users or insert backdoors into a site, and thus be used for site takeover.

Timeline

August 19, 2021 – We finish our investigation and begin the disclosure process for NextScripts: Social Networks Auto-Poster.
August 20, 2021 – We send over full disclosure to the plugin developer.
September 27, 2021 – We follow up with the plugin developer as the plugin has not yet been patched.
October 4, 2021 – A patched version of the plugin, 4.3.21, becomes available.

Conclusion

In today's post, we covered a reflected Cross-Site Scripting(XSS) vulnerability that relied on a relatively obscure quirk of how PHP handles superglobal variables.

All Wordfence users, including Wordfence Premium customers and free Wordfence users are protected by our firewall's built-in XSS protection. Nonetheless, we strongly recommend updating to the latest version of the plugin available

If you know a friend or colleague who is using this plugin on their site, we highly recommend forwarding this advisory to them to help keep their sites protected as this is a serious vulnerability that can lead to complete site takeover.

If your site has been compromised by an attack on this or any other plugin, ourProfessional Site Cleaning services can help you get back in business.

Did you enjoy this post? Share it!

Comments

No Comments

Our business hours are 9am-8pm ET, 6am-5pm PT and 2pm-1am UTC/GMT excluding weekends and holidays.
Response customers receive 24-hour support, 365 days a year, with a 1-hour response time.

Terms of Service Privacy Policy

CCPA Privacy Notice

Products

- Wordfence Free
- Wordfence Premium
- Wordfence Care
- Wordfence Response
- Wordfence Central

Support

- Documentation
- Learning Center
- Free Support
- Premium Support

News

- Blog
- In The News
- Vulnerability Advisories

About

- About Wordfence
- Careers
- Contact
- Security
- CVE Request Form