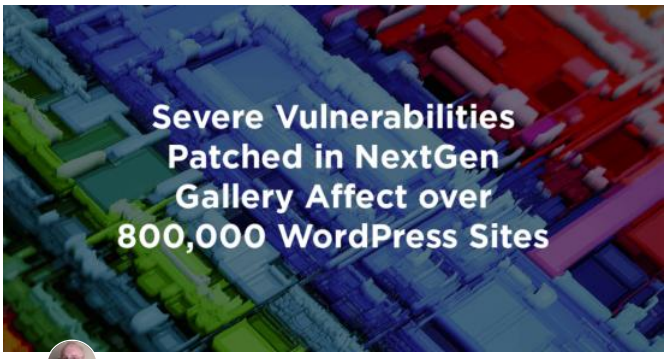

Ram Gall

February 8, 2021

Severe Vulnerabilities Patched in NextGen Gallery Affect over 800,000 WordPress Sites

On December 14, 2020, the Wordfence Threat Intelligence team finished researching two Cross-Site Request Forgery (CSRF) vulnerabilities in NextGen Gallery, a WordPress plugin with over 800,000 installations, including a critical severity vulnerability that could lead to Remote Code Execution(RCE) and Stored Cross-Site Scripting(XSS). Exploitation of these vulnerabilities could lead to a site takeover, malicious redirects, spam injection, phishing, and much more.

We initially reached out to the plugin's publisher, Imagely, the same day, and provided full disclosure the next day, on December 15, 2020. Imagely sent us patches for review on December 16, and published the patched version, 3.5.0, on December 17, 2020.

Wordfence Premium users received firewall rules protecting against these vulnerabilities on December 14, 2020. Sites still running the free version of Wordfence received these rules 30 days later, on January 13, 2021.

Description: Cross-Site Request Forgery (CSRF) leading to XSS and RCE via file upload and LFI
Affected Plugin: WordPress Gallery Plugin – NextGEN Gallery
Plugin Slug: nextgen-gallery
Affected Versions: < 3.5.0
CVE ID: CVE-2020-35942
CVSS Score: 9.6 (CRITICAL)
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H
Fully Patched Version: 3.5.0

NextGEN Gallery is a popular WordPress plugin designed to create highly responsive image galleries. It is clear the plugin's developer took care to integrate security in the code of the plugin. NextGen Gallery has a single security function, `is_authorized_request`, that is used to protect most of its settings:

```
function is_authorized_request($privilege = NULL)
{
    $retval = TRUE;
    if (!$privilege) {
        $privilege = $this->object->get_required_permission();
    }
    // Ensure that the user has permission to access this page
    if (!M_Security::is_allowed($privilege)) {
        $retval = FALSE;
    }
    // Ensure that nonce is valid
    if ($this->object->is_post_request() && (isset($_REQUEST['nonce']) && !M_Security::verify_nonce($_REQUEST['nonce
        $retval = FALSE;
    }
    return $retval;
}
```

This function integrated both a capability check and a nonce check into a single function for easier application throughout the plugin. Unfortunately, a logic flaw in the `is_authorized_request` function meant that the nonce check would allow requests to proceed if the `$_REQUEST['nonce']` parameter was missing, rather than invalid.

This opened up a number of opportunities for attackers to exploit via Cross-Site Request Forgery. One feature of NextGen Gallery is the ability for administrators to upload custom CSS files to be used to style galleries. While the file uploaded had to end with the .css extension, it was possible to upload arbitrary code with double extensions, (e.g., `file.php.css`). While these files would only be executable on certain configurations, such as Apache/mod_php with an AddHandler directive, this could still result in remote code execution on any vulnerable configurations.

Unfortunately, it was also possible to achieve code execution even on configurations not vulnerable to double extensions. NextGen Gallery has a separate feature that allows users to specify how galleries are viewed via a "Legacy Templates" feature, which also uses the `is_authorized_request` function for security. Thus, it was possible to set various album types to use a template with the absolute path of the file uploaded in the previous step, or perform a directory traversal attack using the relative path of the uploaded file, regardless of that file's extension, through a CSRF attack.

This would result in Local File Inclusion (LFI) and Remote code Execution (RCE), as the uploaded file would then be included and executed whenever the selected album type was viewed on the site. Any JavaScript included in the uploaded file would also be executed, resulting in Cross-Site Scripting (XSS).

As a reminder, once an attacker achieves Remote Code Execution on a website, they have effectively taken over that site. XSS can likewise be used to take over a site if a logged-in administrator visits a page running a malicious injected script.

This attack would likely require some degree of social engineering, as an attacker would have to trick an administrator into clicking a link that submitted crafted requests to perform these actions. Additionally, performing these actions would require 2 separate requests, though this would be trivial to implement and we were able to do so during our testing. Finally, the site would require at least one album to be published and accessible to the attacker.

Description: Cross-Site Request Forgery (CSRF) leading to file upload
Affected Plugin: WordPress Gallery Plugin – NextGEN Gallery
Plugin Slug: nextgen-gallery
Affected Versions: < 3.5.0
CVE ID: CVE-2020-35943
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
Fully Patched Version: 3.5.0

NextGen Gallery also used a separate security function, `validate_ajax_request`, for various AJAX actions including those used to upload images:

```
function validate_ajax_request($action = NULL, $token = false)
{
    return (!$action || M_Security::is_allowed($action)) && (!$token || M_Security::verify_nonce($token, $action));
}
```

This function had a similar logic flaw that would allow requests to proceed if the `$_REQUEST['nonce']` parameter was missing, rather than invalid.

This made it possible to trick an administrator into submitting a request crafted to upload an arbitrary image file. While the uploaded file had to be a valid image file, it is possible to hide a webshell or other executable PHP code within such an image file.

This could also be combined with the previous vulnerability, and the image file could be set as a "Legacy Template", at which point it would be included and the code within would be executed. Again, this would require some degree of social engineering, as an attacker would have to trick an administrator into clicking a link that resulted in these requests being sent.

Timeline

December 14, 2020 – The Wordfence Threat Intelligence team finishes researching vulnerabilities in NextGen Gallery. We deploy firewall rules and reach out to Imagely.
December 15, 2020 – Imagely replies and we provide full disclosure.
December 16, 2020 – Imagely sends us a patched version of the plugin to review.
December 17, 2020 – A patched version of NextGen Gallery is made available to the public.
January 13, 2021 – Sites running the free version of Wordfence receive firewall rules.

Conclusion

In today's post, we covered two vulnerabilities in NextGen Gallery, including a Critical Severity Cross-Site Request Forgery (CSRF) that could be used to take over a site via Remote Code Execution (RCE). These vulnerabilities have been fully patched in version 3.5.0, and we strongly recommend that site owners immediately update to the latest version available at this time, which is 3.5.0.

Wordfence Premium users received firewall rules protecting against these vulnerabilities on December 14, 2020. Sites still running the free version of Wordfence received these rules on January 13, 2021.

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 these are critical and high severity vulnerabilities that can lead to full site takeover.

Special thanks to Threat Analyst Chloe Chamberland, who helped analyze this vulnerability, as well as to the plugin's publisher, Imagely, for their fast and professional response.

Comments

4 Comments

Simon *
February 8, 2021
10:53 am

Thanks Wordfence. My goodness Wordpress and its plugins are certainly keeping you busy.

I for one, am moving away from Wordpress for new website projects. Its many vulnerabilities make it too stressful, time consuming and costly to offer to businesses as a professional CMS.

`si *
February 8, 2021
5:30 pm

Thank you for this update.

Nick *
February 10, 2021
8:43 am

Thanks for the update. Does this mean that with the updated rules site is protected from this vulnerability even if not updated? Because I have an older version and if I update the plugin it breaks some app on the front.

Ram Gall *
February 10, 2021
1:28 pm

Hi Nick,

Our firewall rules are able to protect you from these particular vulnerabilities, but we still strongly recommend finding a way to update the plugin as soon as possible. Using older versions of plugins will only get more and more dangerous as new vulnerabilities are discovered.

Stay Updated

Sign up for news and updates from our panel of experienced security professionals.

you@example.com

By checking this box I agree to the terms of service and privacy policy.*

SIGN UP