SecureIT maverick

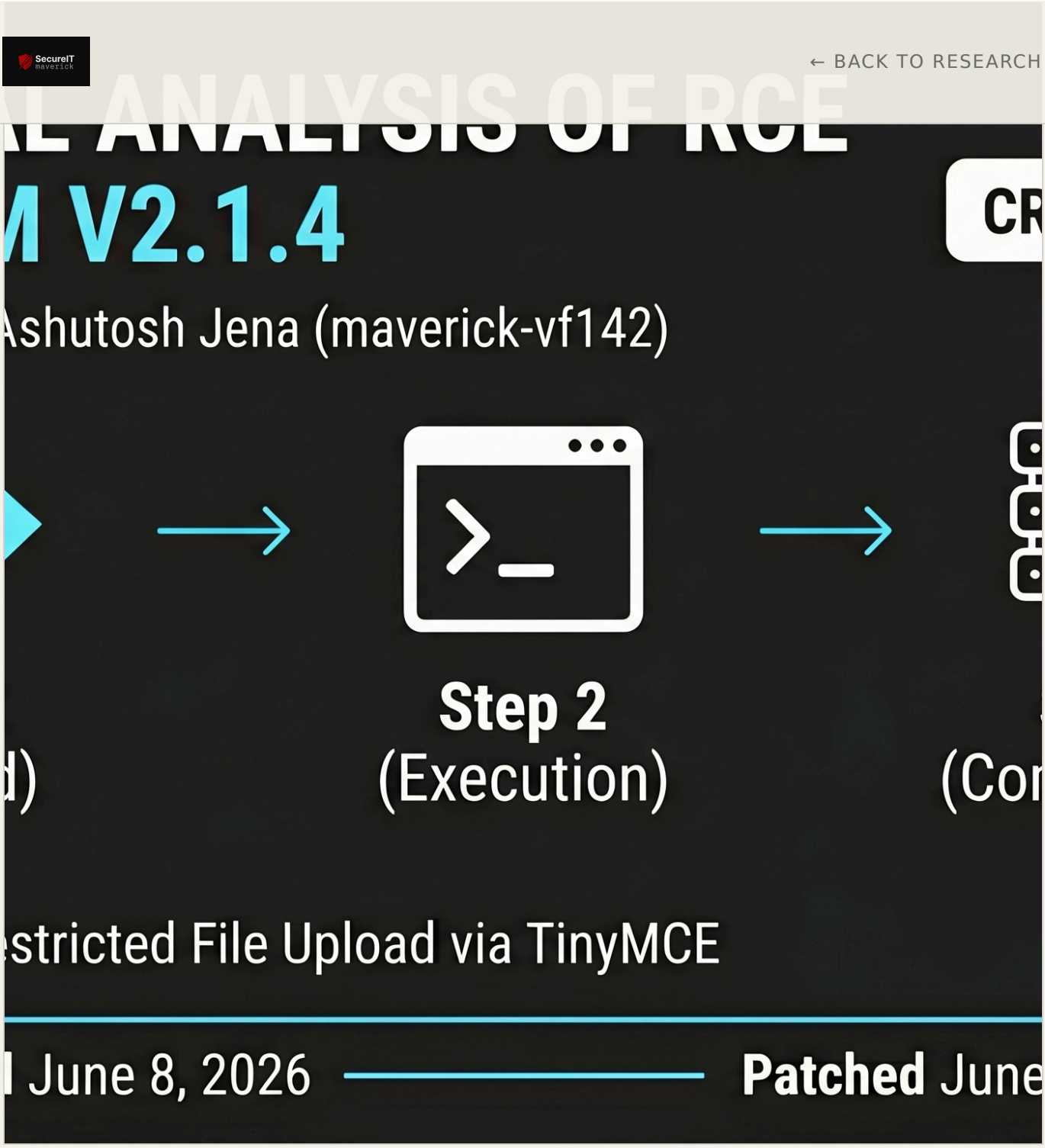
← BACK TO RESEARCH

— VULN RESEARCH

Remote Code Execution (RCE) via Unrestricted File Upload in UnoPim v2.1.4

Executive Summary During a security assessment of UnoPim v2.1.4, I identified a critical Remote Code Execution (RCE) vulnerability. The flaw stems from an unrestricted file upload mechanism within th

2026-06-11 · 3 min read



Executive Summary

During a security assessment of UnoPim v2.1.4, I identified a critical Remote Code Execution (RCE) vulnerability. The flaw stems from an unrestricted file upload mechanism within the TinyMCE editor, which fails to adequately

validate file types. An authenticated administrative user can exploit this to upload and execute arbitrary PHP code on the server, leading to a full system compromise.

Vulnerability Type: Unrestricted Upload of File with Dangerous Type (CWE434)

Severity: Critical (CVSS 3.1 Score: 9.1)

Affected Component:
packages/Webkul/Admin/src/Http/Controllers/TinyMCEController.php

Vulnerability Status
Affected Version: UnoPim v2.1.4 and earlier.

Fixed Version: This vulnerability has been confirmed as remediated in UnoPim v2.1.5.

Note: Users are strongly advised to update to version 2.1.5 or higher to mitigate this risk.

Technical Description

The application provides a file upload endpoint at POST /admin/tinymce/upload for users to insert images into content. While the FileStorer helper includes a Sanitizer trait for SVG files, all other file types are stored without rigorous extension or MIME type validation.

Consequently, a malicious user can upload executable scripts (such as .php files) that are stored directly on the public storage disk. Because these files are immediately webaccessible via the URL returned in the server's response, an attacker can trigger the execution of arbitrary operating system commands.

Proof of Concept (PoC)

Prerequisites

Authenticated access to the administrative dashboard.

An interception proxy (e.g., Burp Suite).

Reproduction Steps

1.
Authenticate: Log in to the UnoPim administrative panel.

2.
Navigate: Go to any page featuring the TinyMCE editor (e.g., Catalog Categories Edit).

3.
Upload: Select the image upload button in the toolbar and prepare a file for upload.

4.
Intercept: Capture the multipart POST request sent to /admin/tinymce/upload using your proxy.

5.

Payload: Modify the request as follows:

Change the filename from the legitimate image name to RCE.php.

Replace the image binary body with the following PHP web shell:

6. Verify: Forward the request. The server will return a 200 OK status and the direct path to your uploaded shell.

7.
Execute: Visit the returned URL with the c parameter to execute commands, for example: http://<host/storage/tinymce/RCE.php?c=id.

Impact

A successful exploit allows an authenticated administrator to achieve:

Full remote code execution on the server.

Unauthorized access to sensitive files, including the .env file (containing database credentials and the APPKEY).

Privilege escalation via group memberships (e.g., lxd or docker).

Full system compromise.

Timeline of Disclosure

June 8, 2026: Vulnerability identified and reported to the Webkul security team.

June 9, 2026: Acknowledgment received from the vendor.

June 10, 2026: Webkul informed me that the issue had been identified internally through their own security assessment and addressed in the latest version of the application. They declined to assign a CVE or provide formal credit.

Remediation Recommendations

To secure the application, I recommended the following measures:

1.
Strict Validation: Implement robust serverside validation to allow only intended file types (e.g., jpg, jpeg, png, gif, webp, pdf).

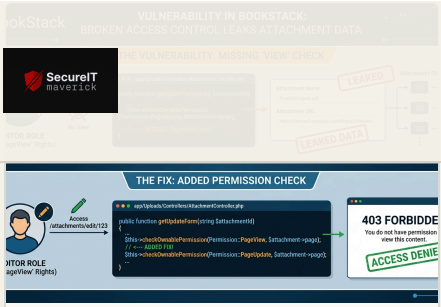
2.
Storage Security: Store uploaded files outside of the web root or on a nonexecutable storage disk (e.g., S3).

3.
File Renaming: Always rename uploaded files to a cryptographically secure random UUID to prevent predictable execution paths.

RELATED POSTS

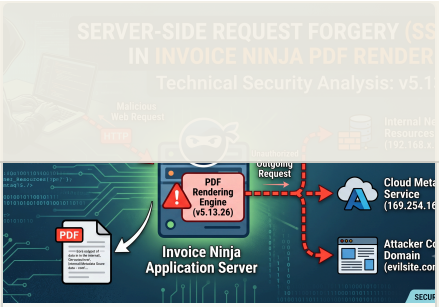
SecureIT maverick

← BACK TO RESEARCH



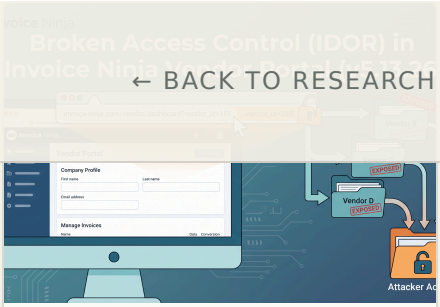
VULN RESEARCH

Broken Access Control in BookStack: How a Missing Permission Check Leaked Attachment Data



VULN RESEARCH

Server-Side Request Forgery (SSRF) in Invoice Ninja PDF Rendering (v5.13.26)



VULN RESEARCH

Broken Access Control (IDOR) in Invoice Ninja Vendor Portal (v5.13.26)

Continue the discussion on Discord

Ask questions, share writeups, talk security with the community

Join →

DISCUSSION

Name

Email (not published)

Share your thoughts, ask a question, or point out a typo…

Comments are moderated and appear after approval.

POST COMMENT

Get new write-ups in your inbox

No spam. Just CTF writeups, vuln research, and guide updates when they drop.

you@example.com SUBSCRIBE

← BACK TO RESEARCH

COPY LINK ← BACK TO RESEARCH

© 2026 Ashutosh Jena · MAVERICK-VF142