INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-30 14:08 UTC

WPFunnels Mail Mint WordPress Plugin - Exposure of Sensitive Information to an Unauthorized Actor

CVE VULNERABILITY | HIGH | CVSS 7.5 | CISA KEV

SCC Item ID	SCC-CVE-2026-0099
Type	CVE Vulnerability
CVE ID	CVE-2026-2025
Severity	HIGH
CVSS Base Score	7.5
EPSS Score	0.3396 (97th percentile)
KEV Status	Yes — CISA Known Exploited Vulnerability
Affected Products	Mail Mint WordPress Plugin before version 1.19.5
Published	2026-04-30T00:00:00Z
Discovery Source	Vulncheck Kev

Executive Summary

A missing authorization control in the Mail Mint WordPress plugin (before version 1.19.5) allows any unauthenticated user on the internet to retrieve the email addresses of every registered user on an affected WordPress site. This vulnerability is actively exploited in the wild, confirmed by listings in both the CISA and VulnCheck Known Exploited Vulnerabilities catalogs. Organizations running affected versions face immediate risk of user email enumeration, which enables targeted phishing, credential stuffing, and regulatory exposure under data protection frameworks.

Technical Analysis

CVE-2026-2025 affects the WPFunnels Mail Mint WordPress plugin before version 1.19.5. The vulnerability stems from a missing authorization check (CWE-862) on at least one REST API endpoint, resulting in unauthorized exposure of sensitive information (CWE-200). An unauthenticated remote attacker can send a crafted HTTP request to the unprotected REST endpoint and receive a full enumeration of registered user email addresses without any authentication token, session, or privilege. CVSS base score: 7.5 (High). EPSS score: 0.33956 (96.98th percentile), indicating a very high probability of active exploitation relative to the broader CVE population. Active exploitation confirmed via CISA KEV and VulnCheck KEV catalog listings. MITRE techniques:

T1589.002 (Gather Victim Identity Information: Email Addresses). Remediation: upgrade to Mail Mint version 1.19.5 or later. Source: CISA KEV (https://www.cisa.gov/known-exploited-vulnerabilities-catalog), NVD (https://nvd.nist.gov/vuln/detail/CVE-2026-2025).

Action Checklist

1. Immediately identify all WordPress installations running Mail Mint before version 1.19.5. If patching cannot be completed within 24 hours, restrict access to the WordPress REST API at the web server or WAF layer for unauthenticated requests, or place the affected sites behind authenticated access controls until patching is complete.
2. Query web server and WordPress access logs for unauthenticated REST API requests targeting Mail Mint endpoints (look for requests to /wp-json/ paths associated with Mail Mint or WPFunnels that return large JSON payloads to unauthenticated sources). Correlate source IPs against threat intelligence feeds. High-frequency requests from a single IP to these endpoints are a strong exploitation indicator.
3. Update Mail Mint to version 1.19.5 or later via the WordPress plugin dashboard or manual upload. Confirm the update resolves the authorization control gap by testing the previously vulnerable REST endpoint post-patch to verify it returns a 401 or 403 for unauthenticated requests.
4. After patching, audit exported or logged REST API responses during the exposure window to determine whether email enumeration occurred. Notify affected users if evidence of enumeration exists, consistent with applicable breach notification obligations. Resume normal WAF/IPS posture once patch is verified.
5. Review all other WordPress plugins and REST API endpoints for missing authorization controls; implement a policy requiring authentication for all non-public REST API endpoints. Consider deploying a WordPress security scanner (e.g., WPScan) as a recurring control. Map this finding to CWE-862 and assess whether other plugins in your inventory share the same weakness class.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to legal, privacy counsel, and executive leadership if access log analysis confirms HTTP 200 responses containing WordPress user email data were returned to unauthenticated external IPs, as this constitutes a confirmed PII exposure event that may trigger GDPR Article 33, CCPA, or US state breach notification obligations depending on jurisdiction and user base.
Recovery Notes	After patching Mail Mint to 1.19.5 and verifying the REST endpoint returns 401/403 for unauthenticated requests, monitor web server access logs for at least 30 days for continued probing of /wp-json/mail-mint/ and /wp-json/wpfunnels/ paths from the same source IPs identified during the exposure window, as threat actors who successfully enumerated the list may return to test other attack surfaces or confirm patch status. Validate that no rogue WordPress admin accounts were created during the exposure window by auditing 'wp user list --role=administrator' and reviewing wp_users table entries created between the exposure start date and patch date. Confirm WAF rules or REST API restrictions added during containment are fully removed or superseded by the patch, and that no legitimate Mail Mint functionality (form submissions, automation triggers) was broken by the containment measures.

Forensic Artifacts	Web server access logs (/var/log/nginx/access.log or /var/log/apache2/access.log): Filter for GET requests to /wp-json/mail-mint/ or /wp-json/wpfunnels/ returning HTTP 200 with response body size >1KB to source IPs presenting no WordPress authentication cookie — this is the primary artifact confirming exploitation of CVE-2026-2025. | WordPress database wp_users table export: A timestamped dump ('wp db export --tables=wp_users backup.sql') establishes the full scope of email addresses exposed and the maximum enumeration blast radius achievable by an attacker who successfully called the vulnerable endpoint. | Mail Mint plugin REST route registration file (pre-patch): The PHP file under /wp-content/plugins/mail-mint/ that registers the vulnerable REST route — specifically the 'permission_callback' definition (or absence thereof) — is the direct code artifact proving the CWE-862 condition existed and is preserved for post-incident documentation and insurance/legal purposes. | Wordfence or ModSecurity WAF logs: If Wordfence (free) or ModSecurity is active, its Live Traffic or audit log will contain the full request URI, source IP, user-agent, and timestamp for REST API hits against Mail Mint endpoints, providing higher-fidelity evidence than raw access logs and potentially capturing blocked exploitation attempts that preceded successful ones. | SMTP/outbound mail server logs for the post-exposure period: If the enumerated email list was weaponized for downstream phishing, the organization's mail gateway or hosted SMTP provider logs (e.g., SendGrid, Mailgun activity logs, or local Postfix /var/log/mail.log) may show a spike in outbound or inbound phishing targeting the exposed addresses, linking the enumeration event to confirmed downstream abuse.

Per-Action IR Details

Containment — Immediately identify all WordPress installations running Mail Mint before version 1.19.5. If patching cannot be completed within 24 hours, restrict access to the WordPress REST API at the web server or WAF layer for unauthenticated requests, or place the affected sites behind authenticated access controls until patching is complete.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST AC-3 (Access Enforcement), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Run 'wp plugin list --format=json | grep mail-mint' via WP-CLI across all managed WordPress installs to enumerate affected versions. If WAF is unavailable, add a Nginx location block or Apache .htaccess rule to return 403 on unauthenticated requests matching '^/wp-json/mail-mint/' and '^/wp-json/wpfunnels/'. On shared hosts without server config access, install and activate the free 'Disable REST API' plugin as a temporary block, or add 'add_filter("rest_authentication_errors", ...)' logic via a must-use plugin to require authentication on all Mail Mint namespace routes.

Evidence: Before restricting access, capture a snapshot of current Apache/Nginx access logs (typically /var/log/apache2/access.log or /var/log/nginx/access.log) filtered for '/wp-json/' requests with HTTP 200 responses returned to unauthenticated clients (no Authorization header, no WordPress auth cookie). Preserve these logs as read-only copies with SHA-256 checksums before any WAF rule changes overwrite or rotate them. Also capture WordPress option table values ('wp_options' rows for mail-mint and wpfunnels plugin versions) via 'wp option list --search="mail_mint*"' to document the vulnerable state at time of containment.

Detection — Query web server and WordPress access logs for unauthenticated REST API requests targeting Mail Mint endpoints (look for requests to /wp-json/ paths associated with Mail Mint or WPFunnels that return large JSON payloads to unauthenticated sources). Correlate source IPs against threat intelligence feeds. High-frequency requests from a single IP to these endpoints are a strong exploitation indicator.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Use the following bash one-liner against your access log to surface exploitation attempts: grep -E 'GET /wp-json/(mail-mint|wpfunnels|mailmint)' /var/log/nginx/access.log | awk '{print $1, $7, $9, $10}' | grep ' 200 ' | sort | uniq -c | sort -rn | head -50. Responses with byte-size fields above 5000 bytes (column 10) to unauthenticated sources are high-confidence exploitation indicators. For IP threat correlation without a SIEM, pipe extracted source IPs into AbuseIPDB's free API: while read ip; do curl -s "https://api.abuseipdb.com/api/v2/check?ipAddress=$ip&maxAgeInDays=90" -H "Key: YOUR_API_KEY" | jq '.data.abuseConfidenceScore'; done < ips.txt. Use Sigma rule 'web_cve_2026_2025_mail_mint_enum' (or author one from scratch) targeting the REST namespace pattern.

Evidence: Primary artifact: web server access logs showing GET requests to /wp-json/mail-mint/v1/contacts or equivalent subscriber-list endpoint, returning HTTP 200 with JSON bodies containing 'email' fields, sourced from IPs with no WordPress session cookie. Secondary artifact: WordPress debug log (/wp-content/debug.log if WP_DEBUG_LOG is enabled) may contain REST API dispatch traces. Tertiary artifact: WAF logs (if ModSecurity or Cloudflare is in use) for rule hits or anomaly scores against the same URI pattern. Capture response body sizes from access logs — a single request returning >10KB of JSON to an unauthenticated IP is a near-certain enumeration event.

Eradication — Update Mail Mint to version 1.19.5 or later via the WordPress plugin dashboard or manual upload. Confirm the update resolves the authorization control gap by testing the previously vulnerable REST endpoint post-patch to verify it returns a 401 or 403 for unauthenticated requests.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-3 (Configuration Change Control), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: If the WordPress admin dashboard is unavailable, download Mail Mint 1.19.5 from wordpress.org/plugins/mail-mint, extract the zip, and replace /wp-content/plugins/mail-mint/ via SFTP, then verify file integrity with: md5sum /wp-content/plugins/mail-mint/includes/API/*.php and compare against the checksums published in the plugin's official SVN tag for 1.19.5. Post-patch verification command (no tools required): curl -s -o /dev/null -w '%{http_code}' -X GET 'https://yoursite.com/wp-json/mail-mint/v1/contacts' — a 401 or 403 confirms the authorization check is enforced. If you receive a 200 with subscriber data, the patch has not applied correctly or a caching layer is serving stale responses.

Evidence: Before applying the patch, preserve a file-level hash of the vulnerable plugin files — specifically the REST API route registration file (typically /wp-content/plugins/mail-mint/app/API/Routes/ or similar path containing 'permission_callback' definitions) using: find /wp-content/plugins/mail-mint -name '*.php' -exec md5sum {} \; > mail-mint-pre-patch-hashes.txt. This establishes a forensic baseline confirming the vulnerable code state and supports post-patch diff analysis to verify that the authorization check was actually added in the permission_callback for the affected route.

Recovery — After patching, audit exported or logged REST API responses during the exposure window to determine whether email enumeration occurred. Notify affected users if evidence of enumeration exists, consistent with applicable breach notification obligations. Resume normal WAF/IPS posture once patch is verified.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST IR-6 (Incident Reporting), NIST AU-11 (Audit Record Retention), NIST SI-4 (System Monitoring), CIS 3.4 (Enforce Data Retention)

Compensating: To determine enumeration scope, extract all HTTP 200 responses to /wp-json/mail-mint/ (or WPFunnels equivalent) endpoints and sum the response byte sizes across the exposure window: awk

'/\/wp-json\/mail-mint/ && $9=="200" {sum+=$10; count++} END {print count" requests, "sum" bytes transferred"}' /var/log/nginx/access.log. Cross-reference the total registered user count ('wp user list --format=count') against the byte volume to estimate whether a full enumeration occurred. If your WordPress site uses Wordfence (free tier), run a scan post-patch and review Wordfence's Live Traffic log for the same endpoint pattern, which retains the last 30 days of requests with IP and user-agent detail.

Evidence: The key recovery-phase artifact is the totality of HTTP 200 responses returned to unauthenticated IPs from the Mail Mint REST endpoint during the exposure window — specifically: (1) access log entries with response sizes, timestamps, and source IPs; (2) WordPress user table row count at time of incident ('SELECT COUNT(*) FROM wp_users') to establish the maximum enumeration blast radius; (3) any wp_mail or SMTP server logs that might show outbound phishing originating from the enumerated list post-exploitation, confirming downstream abuse. Preserve all artifacts under legal hold if breach notification thresholds are met.

Post-Incident — Review all other WordPress plugins and REST API endpoints for missing authorization controls; implement a policy requiring authentication for all non-public REST API endpoints. Consider deploying a WordPress security scanner (e.g., WPScan) as a recurring control. Map this finding to CWE-862 and assess whether other plugins in your inventory share the same weakness class.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST IR-4 (Incident Handling), NIST RA-3 (Risk Assessment), NIST SI-2 (Flaw Remediation), NIST CA-7 (Continuous Monitoring), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run 'wpscan --url https://yoursite.com --enumerate p --plugins-detection aggressive --api-token YOUR_TOKEN' monthly to flag plugins with known CVEs, specifically filtering output for CWE-862 (Missing Authorization) findings. To audit REST API permission_callback coverage across all active plugins without a paid tool, run: grep -rn 'permission_callback' /wp-content/plugins/*/includes/ and flag any routes where the callback returns '__return_true' or is omitted entirely — these are CWE-862 candidates. Add this grep check to your CI/CD pipeline or a weekly cron job. Document results in a plugin risk register mapped to CWE class.

Evidence: Post-incident artifacts for lessons-learned documentation: (1) complete WPScan output enumerating all REST API-exposing plugins and their current CVE status; (2) the grep output identifying all permission_callback definitions across installed plugins, annotated for missing or permissive callbacks; (3) the timeline of exposure window (plugin installation or last known-good version date through patch date), derived from WordPress update history ('wp plugin list --format=json' combined with server file modification timestamps on plugin directories); (4) the incident timeline and response metrics (MTTD, MTTR) for inclusion in the post-incident report per NIST 800-61r3 §4.

Detection Guidance

Review web server access logs and WordPress debug logs for unauthenticated GET or POST requests to /wp-json/ paths associated with Mail Mint or WPFunnels. Exploitation signatures include high-volume requests from a single IP, requests that return HTTP 200 with a JSON body containing multiple email address values, and requests with no Authorization header or WordPress authentication cookie. If a WAF is deployed, create a rule alerting on unauthenticated requests to Mail Mint REST API paths. No public IOC hashes or specific endpoint paths have been confirmed in available sources at this time; monitor CISA KEV and VulnCheck for updated indicators.

Framework Mappings

MITRE-ATTACK

- **T1589.002** — Email Addresses

- **T1590** — Gather Victim Network Information

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

NIST-800-53R5

- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest
- **IR-5** — Incident Monitoring

CIS-V8

- **6.1** — Establish an Access Granting Process

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information

NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1589.002	Email Addresses	Reconnaissance
T1590	Gather Victim Network Information	Reconnaissance

Sources

Source	URL	Tier
vulncheck_kev	https://nvd.nist.gov/vuln/detail/CVE-2026-2025	T1
Known Exploited Vulnerabilities Catalog	https://www.cisa.gov/known-exploited-vulnerabilities-catalog	T1
CVE Record: CVE-2025-2026	https://www.cve.org/CVERecord?id=CVE-2025-2026	T3
March 2026 CVE Landscape: 31 High-Impact ...	https://www.recordedfuture.com/blog/march-2026-cve-landscape	T3
CVE-2026-20025 - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-20025	T1

DISCLAIMER

Generated 2026-04-30 14:08 UTC by TJS Security Command Center