INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-20 06:54 UTC

Gravity SMTP Plugin Exposes Email Credentials at Scale: 17M Exploitation Attempts Signal Broad Unpatched Attack Surface

CVE VULNERABILITY | HIGH | CVSS 7.5

SCC Item ID	SCC-CVE-2026-0339
Type	CVE Vulnerability
CVE ID	CVE-2026-4020, CVE-2026-8713
Severity	HIGH
CVSS Base Score	7.5
EPSS Score	0.0298 (86th percentile)
Affected Products	Gravity SMTP WordPress plugin versions 2.1.4 and earlier (patched in 2.1.5, released March 17); connected email services include Amazon SES, Google OAuth, Mailjet, Resend, and Zoho
Published	2026-06-19T16:25:02
Discovery Source	Rss

Executive Summary

A high-severity unauthenticated vulnerability in the Gravity SMTP WordPress plugin (CVE-2026-4020) exposes live email service credentials, including API keys and OAuth tokens for Amazon SES, Google, Mailjet, Resend, and Zoho, to any unauthenticated attacker who queries a single REST API endpoint. Approximately 100,000 active installations are affected, and exploitation has exceeded 17 million blocked attempts, with a single-day spike of 4 million requests on June 7, indicating active, widespread scanning. Organizations still running version 2.1.4 or earlier face immediate risk of email infrastructure takeover, bulk spam abuse, sensitive data exfiltration, and downstream account compromise across connected services.

Technical Analysis

CVE-2026-4020 affects Gravity SMTP plugin versions 2.1.4 and earlier for WordPress. The vulnerability resides in a permissive REST API endpoint that returns a full system diagnostic report, including live API keys, OAuth tokens, and email service credentials for Amazon SES, Google, Mailjet, Resend, and Zoho, to any unauthenticated requester without access control enforcement. Applicable CWEs include CWE-306 (Missing

Authentication for Critical Function), CWE-862 (Missing Authorization), and CWE-200 (Exposure of Sensitive Information to an Unauthorized Actor). CVSS base score is 7.5 (High). EPSS score is 0.0298 at the 85.5th percentile, indicating elevated exploitation probability relative to the broader CVE population. MITRE ATT&CK techniques include T1552 (Unsecured Credentials), T1552.001 (Credentials In Files), T1190 (Exploit Public-Facing Application), T1078.001 (Default Accounts), T1592.002 (Gather Victim Host Information), and T1566 (Phishing). The patch was released March 17 as version 2.1.5. The June 7 spike confirms a large proportion of installations remain unpatched more than two months after patch availability. A secondary CVE, CVE-2026-8713, is associated with the Avada Builder plugin (pre-3.15.4); technical details are not fully verified from authoritative sources and are treated at lower confidence in this report and are recommended for separate advisory coverage.

Action Checklist

1. Step 1: Containment. Immediately identify all WordPress installations running Gravity SMTP 2.1.4 or earlier across your environment. Prioritize patching to version 2.1.5 within 24 hours. While patch deployment is in progress, block unauthenticated external access to the plugin's REST API endpoint at the WAF or perimeter layer. If the WordPress REST API is not required to be public-facing, restrict it via firewall rules or server configuration.
2. Step 2: Detection. Query WordPress access logs and WAF logs for unauthenticated GET requests to the Gravity SMTP diagnostic REST API endpoint. Look for high-frequency requests from a single IP or distributed scanning patterns targeting that endpoint. Review SIEM for anomalous outbound email volume via Amazon SES, Mailjet, Resend, or Zoho that may indicate credential abuse post-exfiltration. NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs) apply here.
3. Step 3: Eradication. Update all Gravity SMTP plugin installations to version 2.1.5 or later, released March 17 via the WordPress plugin repository. Immediately rotate all credentials that may have been exposed: Amazon SES API keys, Google OAuth tokens, Mailjet API keys, Resend API keys, and Zoho credentials configured within the plugin. Treat any credential present in the diagnostic report as fully compromised regardless of observed exploitation. NIST AC-2 (Account Management) and NIST IA-4 (Identifier Management) apply.
4. Step 4: Recovery. After patching, verify the diagnostic REST API endpoint no longer returns credential data to unauthenticated requests. Monitor connected email service accounts (Amazon SES, Google, Mailjet, Resend, Zoho) for unauthorized send activity, API key usage from unexpected IPs, and quota anomalies for a minimum of 30 days post-remediation. Re-enable any WAF rules relaxed during triage only after patch confirmation. CIS 7.1 (Establish and Maintain a Vulnerability Management Process) applies.
5. Step 5: Post-Incident. Conduct an inventory of all WordPress plugins across managed environments using an asset inventory process consistent with CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) and CIS 2.1 (Establish and Maintain a Software Inventory). Evaluate whether any additional plugins expose unauthenticated REST API endpoints returning sensitive configuration data. Implement automated patch management for WordPress plugin updates per CIS 7.4 (Perform Automated Application Patch Management). Review REST API authentication configurations against NIST AC-17 (Remote Access) and AC-6 (Least Privilege) to prevent recurrence of unauthenticated access to sensitive diagnostic functions.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to legal counsel and executive leadership immediately if forensic review of Amazon SES, Google, Mailjet, Resend, or Zoho send logs confirms unauthorized email sends occurred during the credential exposure window, as this constitutes potential unauthorized use of organizational email infrastructure affecting third-party recipients and may trigger breach notification obligations under applicable data protection regulations (e.g., GDPR, CAN-SPAM, state privacy laws) depending on the content and recipients of those sends.
Recovery Notes	After patching to Gravity SMTP 2.1.5 and rotating all exposed credentials, maintain enhanced monitoring of all five connected email service accounts (Amazon SES, Google, Mailjet, Resend, Zoho) for a minimum of 30 days, specifically watching for sends from unexpected source IPs, API key usage anomalies, and quota spikes that may indicate a threat actor retained copies of the exposed credentials and is exploiting them from a delayed or secondary infrastructure. Any newly issued API keys or OAuth tokens for these services should be scoped to the minimum required permissions and restricted to known application server IP ranges where the service provider supports IP allowlisting. Verify that the Gravity SMTP diagnostic REST API endpoint returns 401 or 404 to unauthenticated requests on all patched installations before closing the incident ticket and restoring normal WAF posture.
Forensic Artifacts	Web server access logs (/var/log/nginx/access.log or /var/log/apache2/access.log) filtered for unauthenticated GET requests to the Gravity SMTP REST API namespace (e.g., /wp-json/gsmtp/v1/debug or equivalent diagnostic route), specifically HTTP 200 responses that indicate the endpoint returned credential data rather than an auth error — the 17M exploitation attempt volume and June 7 spike of 4M requests mean these logs are the primary exploitation timeline artifact. | WordPress database wp_options table export containing Gravity SMTP plugin settings (option_name LIKE 'gravitysmtp%'), which stores the plaintext or lightly encoded Amazon SES API keys, Google OAuth client secrets, Mailjet API keys, Resend API keys, and Zoho credentials that CVE-2026-4020 exposed via the unauthenticated diagnostic endpoint. | Amazon SES send statistics and IAM access key last-used metadata (retrievable via 'aws iam get-access-key-last-used --access-key-id '), Mailjet and Resend event API logs, and Zoho Mail audit logs covering the full exposure window — these confirm whether compromised credentials were actively weaponized for spam, phishing, or bulk email campaigns following exfiltration. | WAF request logs (blocked and allowed) covering the Gravity SMTP REST API namespace, with source IP geolocation and user-agent fields preserved — the distributed scanning pattern at 17M+ attempts indicates automated scanner infrastructure, and clustering source IPs or user-agent strings may enable attribution to specific scanning campaigns or threat actor tooling. | Google OAuth token usage history from the Google Cloud Console (API & Services > Credentials > OAuth 2.0 Client IDs > Token Usage) for any OAuth clients configured in Gravity SMTP, identifying whether the exposed Google OAuth tokens were used for authentication attempts from IPs outside the legitimate application server range during the exploitation window.

Per-Action IR Details

Step 1: Containment — Immediately identify all WordPress installations running Gravity SMTP 2.1.4 or earlier across your environment. Block unauthenticated external access to the plugin's REST API endpoint at the WAF or perimeter layer while patch deployment is staged. If the WordPress REST API is not required to be public-facing, restrict it via firewall rules or server configuration.

NIST Phase: Containment

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

Controls: NIST AC-4 (Information Flow Enforcement), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Run 'wp plugin list --fields=name,version --format=csv' via WP-CLI across all managed WordPress installations and grep for 'gravitysmtp' with version <= 2.1.4. Block the specific REST API route at the nginx/Apache layer with a location block denying access to '/wp-json/gsmtp/' or equivalent Gravity SMTP namespace before patch is applied. Use 'iptables -I INPUT -p tcp --dport 443 -m string --string "/wp-json/gsmtp" --algo bm -j DROP' as an emergency perimeter measure on servers without a WAF.

Evidence: Before implementing any WAF block or firewall rule changes, capture a full snapshot of current web server access logs (e.g., '/var/log/nginx/access.log', '/var/log/apache2/access.log') to preserve the pre-containment request history. Record active TCP connections to port 443 via 'netstat -ano' or 'ss -tnp' to identify any in-progress connections to the Gravity SMTP REST API endpoint. Export existing WAF rule sets and current allow-lists before modification so the pre-incident network posture is preserved for forensic review.

Step 2: Detection — Query WordPress access logs and WAF logs for unauthenticated GET requests to the Gravity SMTP diagnostic REST API endpoint. Look for high-frequency requests from a single IP or distributed scanning patterns targeting that endpoint. Review SIEM for anomalous outbound email volume via Amazon SES, Mailjet, Resend, or Zoho that may indicate credential abuse post-exfiltration. NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs) apply here.

NIST Phase: Detection Analysis

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

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-2 (Event Logging), CIS 8.2 (Collect Audit Logs)

Compensating: Use 'grep -E "GET /wp-json/gsmtp" /var/log/nginx/access.log | awk '{print $1}' | sort | uniq -c | sort -rn' to surface high-frequency source IPs probing the Gravity SMTP diagnostic endpoint without authentication (HTTP 200 responses to unauthenticated GET requests are the primary exploitation indicator). For email abuse detection without a SIEM, query Amazon SES send statistics via 'aws ses get-send-statistics --region us-east-1' and compare against baseline; for Mailjet, use 'curl -s -u API_KEY:SECRET https://api.mailjet.com/v3/REST/message?Limit=50' to review recent send activity. Flag any sends originating from IPs or user-agents not matching your application servers.

Evidence: Capture the full web server access log covering the window from June 7 (the 4-million-request spike day) to present, preserving original timestamps and source IPs — this is the primary artifact showing whether your installation was actively probed. Simultaneously export current outbound email send logs from each connected service (Amazon SES CloudWatch metrics, Mailjet event API, Resend dashboard exports, Zoho Mail logs) before any credential rotation, as post-rotation the compromised key's activity history may become inaccessible. If a WAF is in place, export blocked and allowed request logs for the '/wp-json/gsmtp/' namespace before any rule changes overwrite ring-buffer storage.

Step 3: Eradication — Update all Gravity SMTP plugin installations to version 2.1.5 or later, released March 17 via the WordPress plugin repository. Immediately rotate all credentials that may have been exposed: Amazon SES API keys, Google OAuth tokens, Mailjet API keys, Resend API keys, and Zoho credentials configured within the plugin. Treat any credential present in the diagnostic report as fully compromised regardless of observed exploitation. NIST AC-3 (Access Enforcement) and D3-CRO (Credential Rotation) apply.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST AC-3 (Access Enforcement), NIST SI-2 (Flaw Remediation), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Before patching, run 'wp plugin get gravitysmtp --field=version' to confirm the installed version, then execute 'wp plugin update gravitysmtp' via WP-CLI to pull version 2.1.5 from the WordPress plugin repository. Immediately after patch confirmation, revoke and regenerate credentials in each connected service console: for Amazon SES, use 'aws iam delete-access-key --access-key-id ' followed by 'aws iam create-access-key'; for Google

OAuth, revoke the token at console.cloud.google.com/apis/credentials; for Mailjet and Resend, generate new API keys via their respective dashboards and update the Gravity SMTP plugin settings to reference the new keys. After credential rotation, verify the new keys are functional by triggering a test send via Gravity SMTP's built-in diagnostic tool.

Evidence: CRITICAL — before updating the plugin or rotating any credentials, capture a memory dump or at minimum a process listing ('ps aux', 'Get-Process') of the web server process (e.g., php-fpm, apache2) to preserve any in-memory state. Export the current Gravity SMTP plugin configuration from the WordPress database: 'wp option get gravitysmtp_settings --format=json > gravitysmtp_settings_prerotation.json' — this preserves a forensic record of which credentials were exposed in what configuration state. Capture the WordPress 'wp_options' table dump containing Gravity SMTP settings as a timestamped SQL export before any database writes occur during the update process, as the patch may overwrite or clear the diagnostic endpoint configuration.

Step 4: Recovery — After patching, verify the diagnostic REST API endpoint no longer returns credential data to unauthenticated requests. Monitor connected email service accounts (Amazon SES, Google, Mailjet, Resend, Zoho) for unauthorized send activity, API key usage from unexpected IPs, and quota anomalies for a minimum of 30 days post-remediation. Re-enable any WAF rules relaxed during triage only after patch confirmation. CIS 7.1 (Establish and Maintain a Vulnerability Management Process) applies.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), NIST AU-6 (Audit Record Review, Analysis, And Reporting)

Compensating: Verify remediation by issuing an unauthenticated curl request directly to the formerly vulnerable endpoint: 'curl -i -X GET https:///wp-json/gsmtp/v1/debug' (or the equivalent Gravity SMTP diagnostic route) and confirm the response is either 401 Unauthorized or 404 Not Found — a 200 response with credential data confirms the patch failed or was not applied. Set up a daily cron job for 30 days that queries each email service's send API for activity and compares against a known-good baseline, alerting on any sends exceeding a 110% volume threshold or originating from IPs outside your application server range. Re-enable WAF blocking rules for the Gravity SMTP REST API namespace only after the curl verification confirms the endpoint is protected.

Evidence: Before re-enabling relaxed WAF rules, document the patched plugin version via 'wp plugin get gravitysmtp --fields=name,version,status' and archive the output as a timestamped verification artifact. Capture a post-patch snapshot of each connected email service's API key activity log (Amazon SES IAM last-used data, Google OAuth token usage history, Mailjet/Resend API key last-used timestamps) to establish a clean baseline for the 30-day monitoring window — any activity against old rotated keys appearing after rotation indicates the old key was already in use by an attacker at the time of rotation.

Step 5: Post-Incident — Conduct an inventory of all WordPress plugins across managed environments using an asset inventory process consistent with CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) and CIS 2.1 (Establish and Maintain a Software Inventory). Evaluate whether any additional plugins expose unauthenticated REST API endpoints returning sensitive configuration data. Implement automated patch management for WordPress plugin updates per CIS 7.4 (Perform Automated Application Patch Management). Review REST API authentication configurations against NIST AC-17 (Remote Access) and AC-6 (Least Privilege) to prevent recurrence of unauthenticated access to sensitive diagnostic functions.

NIST Phase: Post Incident

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

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 7.4 (Perform Automated Application Patch Management), NIST AC-6 (Least Privilege), NIST AC-17 (Remote Access)

Compensating: Run 'wp plugin list --fields=name,version,status --format=csv --path=/var/www/html' across all WordPress document roots on managed servers and pipe the output into a central CSV for version auditing. Use a Sigma rule targeting web server access logs to detect any WordPress plugin REST API namespace ('/wp-json/')

returning HTTP 200 to unauthenticated GET requests that include credential-pattern strings (API key, token, secret) in the response body — this creates a detection capability for the CVE-2026-4020 class of misconfigured diagnostic endpoints across all plugins, not just Gravity SMTP. Enable WordPress auto-updates for plugins in 'wp-config.php' via 'define("WP_AUTO_UPDATE_CORE", true)' and configure ManageWP or MainWP (free tiers available) for centralized plugin update management across multi-site environments.

Evidence: Produce and archive a post-incident plugin inventory snapshot including all plugin names, versions, active status, and REST API namespace registrations discoverable via 'wp rest api route list --format=json' — this documents the pre-remediation attack surface and serves as the baseline for future audits. Archive all detection artifacts from Steps 1–4 (web server access logs, WAF block logs, email service API activity exports, pre-rotation database dumps) in an immutable evidence store with chain-of-custody documentation per NIST 800-61r3 §4 post-incident reporting requirements. If any email sends occurred via compromised Amazon SES, Mailjet, Resend, or Zoho credentials during the exposure window, preserve the full send logs as these may constitute evidence of unauthorized use of email infrastructure and could trigger downstream incident notifications to recipients.

Detection Guidance

Search WordPress server access logs and WAF event logs for unauthenticated GET requests targeting the Gravity SMTP plugin's diagnostic REST API endpoint. Indicators include high request volume from single or distributed IPs, requests with no authentication headers, and 200-status responses from the diagnostic endpoint path. In your SIEM, build a query correlating successful unauthenticated calls to that endpoint with subsequent anomalous outbound email events across Amazon SES, Mailjet, Resend, or Zoho. Flag any new API key usage or OAuth token activity from IPs not previously associated with your WordPress host. Behavioral indicators of post-exploitation credential abuse include sudden spikes in outbound email volume, bounce rate increases, and email service provider abuse notifications. Apply NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs) to ensure logging is active and retention covers at least the period from the June 7 spike. NIST AU-6 (Audit Record Review) and NIST SI-7 (Software, Firmware, and Information Integrity) are applicable countermeasures for ongoing monitoring of credential and configuration file integrity.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`REST API diagnostic endpoint of Gravity SMTP plugin (specific path not publicly confirmed in verified sources)`	Unauthenticated GET requests to this endpoint return live API keys and OAuth tokens; exact path should be obtained from the plugin's 2.1.5 patch release notes	HIGH

Framework Mappings

MITRE-ATTACK

- **T1552.001** — Credentials In Files
- **T1552** — Unsecured Credentials
- **T1190** — Exploit Public-Facing Application
- **T1078.001** — Default Accounts

- **T1592.002** — Software
- **T1566** — Phishing

NIST-800-53R5

- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection
- **AC-3** — Access Enforcement
- **SI-10** — Information Input Validation
- **IA-2** — Identification and Authentication (Organizational Users)
- **SC-28** — Protection of Information at Rest
- **IR-5** — Incident Monitoring

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A07:2021** — Identification and Authentication Failures

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **16.12** — Implement Code-Level Security Checks
- **6.3** — Require MFA for Externally-Exposed Applications
- **6.1** — Establish an Access Granting Process
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

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
T1552.001	Credentials In Files	Credential-Access
T1552	Unsecured Credentials	Credential-Access
T1190	Exploit Public-Facing Application	Initial-Access
T1078.001	Default Accounts	Defense-Evasion
T1592.002	Software	Reconnaissance
T1566	Phishing	Initial-Access

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/hackers-exploit-info...	T3
CVE-2026-4020 Detail - NVD	https://nvd.nist.gov/vuln/detail/cve-2026-4020	T1
CVE-2026-4020: Gravity SMTP Plugin Information Disclosure	https://www.sentinelone.com/vulnerability-database/cve-2026-4020/	T3
CVE-2026-8713 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-8713	T3
CVE-2026-21713 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-21713	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-4020, CVE-2026-8713	T1
Google Security Advisory	https://chromereleases.googleblog.com/	T1

Generated 2026-06-20 06:54 UTC by TJS Security Command Center