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

CVE-2026-1890: LeadConnector WordPress Plugin Unauthenticated REST API Authorization Bypass

CVE VULNERABILITY | **HIGH** | CVSS 8.2 | **CISA KEV**

SCC Item ID	SCC-CVE-2026-0100
Type	CVE Vulnerability
CVE ID	CVE-2026-1890
Severity	HIGH
CVSS Base Score	8.2
EPSS Score	0.0007 (22th percentile)
KEV Status	Yes — CISA Known Exploited Vulnerability
Affected Products	LeadConnector WordPress Plugin < 3.0.22
Published	2026-04-30T00:00:00Z
Discovery Source	Vulncheck Kev

Executive Summary

A high-severity authorization flaw in the LeadConnector WordPress plugin (versions before 3.0.22) allows any unauthenticated user to call a protected REST API endpoint and overwrite plugin data without credentials. CISA has added this to its Known Exploited Vulnerabilities catalog, indicating active exploitation in the wild. Organizations running LeadConnector on customer-facing WordPress sites face immediate risk of data manipulation, lead record corruption, and potential follow-on compromise of CRM-connected systems.

Technical Analysis

CVE-2026-1890 is a missing authorization vulnerability (CWE-862) in the LeadConnector WordPress plugin affecting all versions before 3.0.22. The plugin exposes a REST API route that does not enforce WordPress capability checks or nonce validation, permitting unauthenticated remote requests to invoke the endpoint and overwrite data stored by the plugin. Exploitation maps to MITRE ATT&CK T1190 (Exploit Public-Facing Application) for initial access and T1565.001 (Stored Data Manipulation) for impact. CVSS base score is 8.2 (High). EPSS score is 0.00072 (21st percentile) as of available data, though CISA KEV listing indicates confirmed active exploitation regardless of EPSS. No vendor CVSS vector has been published; NVD base score

of 8.2 applies. Remediation is a version upgrade to 3.0.22 or later via the WordPress plugin repository. No alternative mitigation is documented in available sources.

Action Checklist

1. Step 1: Containment. Identify all WordPress instances running LeadConnector < 3.0.22. Block unauthenticated external access to /wp-json/ REST API routes for the LeadConnector plugin at your WAF or perimeter firewall until patching is complete. If you cannot patch immediately, disable the plugin to remove the exposed route.
2. Step 2: Detection. Query web server and WAF logs for unauthenticated POST or PUT requests to LeadConnector REST API routes (paths typically matching /wp-json/leadconnector/ or similar plugin-registered namespaces). Look for requests with no Authorization header, no nonce parameter, and HTTP 200 responses. Cross-reference with unexpected changes in plugin-stored data (lead records, form submissions, CRM sync entries).
3. Step 3: Eradication. Upgrade LeadConnector plugin to version 3.0.22 or later through the WordPress plugin administration panel (Plugins section) or via WP-CLI: wp plugin update leadconnector. Confirm the installed version post-update. Remove the plugin entirely if it is not actively required.
4. Step 4: Recovery. After patching, audit LeadConnector data stores for unexpected modifications: review lead records, form submissions, and any CRM-synced data for anomalous entries or overwrites. Restore from backup any data confirmed as tampered. Re-enable normal REST API access and monitor for continued anomalous requests for at least 72 hours post-patch.
5. Step 5: Post-Incident. This vulnerability exposes a control gap in plugin vetting and REST API exposure management. Implement a WordPress plugin approval process requiring security review before deployment. Add REST API route auditing to your web application scanning program. Review all WordPress plugins for unauthenticated API surface using tools such as WPScan against your environment on a scheduled basis.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to legal, privacy counsel, and senior leadership immediately if log analysis confirms that lead records containing PII (names, emails, phone numbers) or CRM-synced data were successfully overwritten or extracted via the unauthenticated LeadConnector REST API endpoint, as this may trigger breach notification obligations under applicable state privacy laws or GDPR; also escalate if evidence of webshell installation or lateral movement beyond the WordPress application layer is detected, indicating the authorization bypass was used as an initial access vector for deeper compromise.

Recovery Notes	After upgrading to LeadConnector 3.0.22, validate that all /wp-json/leadconnector/ endpoints now return HTTP 401 or 403 for unauthenticated POST and PUT requests before re-enabling full external REST API access, using a curl-based verification script from an unauthenticated external IP. Restore any confirmed-tampered lead records, form submissions, and CRM-synced entries from the most recent clean backup taken prior to the earliest exploitation timestamp identified in the log review. Monitor web server and WAF logs continuously for at least 72 hours post-patch for repeat unauthenticated access attempts to LeadConnector REST routes from the source IPs identified during the detection phase, as CISA KEV listing indicates active threat actor tooling that may retry previously successful targets.
Forensic Artifacts	Apache/Nginx access logs: entries showing unauthenticated POST or PUT requests to /wp-json/leadconnector/ URI paths returning HTTP 200, which represent confirmed successful exploitation of CVE-2026-1890 — the absence of an Authorization header and wp_nonce parameter in these requests distinguishes exploitation traffic from legitimate authenticated plugin API calls. | WordPress database dump of LeadConnector tables (wp_leadconnector_leads, wp_leadconnector_forms, and wp_options rows with 'leadconnector' key prefix): diff against the pre-exploitation clean backup to identify specific records overwritten or injected by the attacker via the unauthorized REST API write operations. | WAF block/allow logs filtered on /wp-json/leadconnector/ path: preserves the complete timeline of exploitation attempts including attacker source IPs, request payload sizes, and any WAF rule matches or bypasses — critical for determining exploitation window duration and scope. | File system integrity snapshot of wp-content/plugins/leadconnector/ directory with timestamps and SHA-256 hashes of all plugin files: detects whether the attacker leveraged the authorization bypass to write a webshell or backdoor into the plugin directory as a secondary persistence mechanism beyond data manipulation. | WordPress debug.log and PHP error logs (wp-content/debug.log): may contain exception traces, REST API handler errors, or unexpected output generated during exploitation of the authorization bypass, providing additional technical detail on which specific LeadConnector REST API callback functions were invoked without authorization.

Per-Action IR Details

Step 1: Containment — Immediately identify all WordPress instances running LeadConnector < 3.0.22. Block unauthenticated external access to /wp-json/ REST API routes for the LeadConnector plugin at your WAF or perimeter firewall until patching is complete. If you cannot patch immediately, disable the plugin to remove the exposed route.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolate affected systems and restrict the attack vector to prevent continued exploitation of the unauthenticated LeadConnector REST API endpoint.

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: If no WAF is available, use the WordPress .htaccess file to block the LeadConnector namespace: add 'RewriteRule ^wp-json/leadconnector/ - [F,L]' to block that route at the Apache/Nginx level. Enumerate all WordPress instances using: find /var/www -name 'wp-config.php' | xargs grep -l '' to locate installs, then run 'wp plugin list --path=/path/to/wp --format=table | grep leadconnector' for each instance via WP-CLI to confirm affected versions.

Evidence: Before blocking WAF rules, export and preserve the current WAF or perimeter firewall access logs covering the past 30 days, focusing on all HTTP requests to paths matching /wp-json/leadconnector/ or the plugin-registered REST namespace. Capture a full directory listing and file hash inventory of the LeadConnector plugin directory (typically wp-content/plugins/leadconnector/) using 'sha256sum wp-content/plugins/leadconnector/*' to establish a pre-patch baseline for integrity comparison. Document the currently installed plugin version via WP-CLI: 'wp plugin get leadconnector --fields=version'.

Step 2: Detection — Query web server and WAF logs for unauthenticated POST or PUT requests to LeadConnector REST API routes (paths typically matching /wp-json/leadconnector/ or similar plugin-registered namespaces). Look for requests with no Authorization header, no nonce parameter, and HTTP 200 responses. Cross-reference with unexpected changes in plugin-stored data (lead records, form submissions, CRM sync entries).

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate web server and WAF log evidence to determine whether CVE-2026-1890 has been actively exploited, identify attacker source IPs, and assess the scope of unauthorized REST API calls against the LeadConnector endpoint.

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Parse Apache/Nginx access logs directly with grep/awk: 'grep -E "(POST|PUT).*wp-json/leadconnector" /var/log/nginx/access.log | grep " 200 "' to isolate successful unauthenticated writes. Pipe through 'awk '{print $1}' | sort | uniq -c | sort -rn' to rank source IPs by request volume. For WordPress database analysis, use WP-CLI to query the options table and custom LeadConnector tables for unexpected modifications: 'wp db query "SELECT option_name, option_value, last_updated FROM wp_options WHERE option_name LIKE '%leadconnector%' ORDER BY last_updated DESC LIMIT 50;"'. Use GoAccess (free, real-time log analyzer) for rapid visual triage of the access log timeline correlated to the vulnerability disclosure date.

Evidence: Collect the full Apache or Nginx access log (access.log / access_log) and error log for the WordPress server, preserving entries with timestamps covering at least 90 days prior to detection to identify reconnaissance and exploitation patterns. Export WAF logs filtering on URI path '/wp-json/leadconnector/' with HTTP methods POST and PUT and response codes 200. Dump the WordPress database LeadConnector-specific tables (typically prefixed wp_leadconnector_ or stored in wp_options under leadconnector keys) to capture the current data state and compare against the most recent clean backup to identify unauthorized overwrites of lead records or CRM sync data.

Step 3: Eradication — Upgrade LeadConnector plugin to version 3.0.22 or later through the WordPress admin dashboard (Plugins > Updates) or via WP-CLI: wp plugin update leadconnector. Confirm the installed version post-update. Remove the plugin entirely if it is not actively required.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the vulnerable LeadConnector plugin code path that registered the unauthenticated REST API endpoint, eliminating the attack surface introduced by CVE-2026-1890.

Controls: NIST SI-2 (Flaw Remediation), NIST CM-4 (Impact Analyses), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: If automated update is unavailable, manually download LeadConnector 3.0.22 from the WordPress plugin repository, verify the package checksum against the published hash, and deploy via SFTP to wp-content/plugins/leadconnector/. Confirm the patch resolved the authorization bypass by checking that POST requests to /wp-json/leadconnector/ from an unauthenticated curl session now return HTTP 401 or 403: 'curl -s -o /dev/null -w "%{http_code}" -X POST https://yoursite.com/wp-json/leadconnector/v1/[endpoint]'. Run WP-CLI post-update: 'wp plugin get leadconnector --fields=version,status' to confirm version 3.0.22 or later is active.

Evidence: Before applying the update, capture a forensic copy of the current vulnerable plugin files — specifically the REST API registration code in the LeadConnector plugin (typically includes/api/ or similar directory) — to document the exact code that registered the unauthenticated route. This preserves evidence of the vulnerable state for post-incident review. Record file modification timestamps on the plugin directory using 'find wp-content/plugins/leadconnector -type f -printf "%T+ %p\n" | sort' to detect any attacker-introduced file modifications beyond normal plugin code, which would indicate the vulnerability was used as an initial access vector for webshell or backdoor placement.

Step 4: Recovery — After patching, audit LeadConnector data stores for unexpected modifications: review lead records, form submissions, and any CRM-synced data for anomalous entries or overwrites. Restore from backup any data confirmed as tampered. Re-enable normal REST API access and monitor for continued

anomalous requests for at least 72 hours post-patch.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore integrity of LeadConnector-managed data (lead records, form submissions, CRM sync entries) corrupted or overwritten via the unauthorized REST API calls, and verify that the patched plugin correctly enforces authorization before resuming normal operations.

Controls: NIST IR-4 (Incident Handling), NIST CP-10 (System Recovery and Reconstitution), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Use WP-CLI to export a current snapshot of all LeadConnector data tables: 'wp db export --tables=$(wp db tables --all-tables | grep leadconnector | tr '\n' ',') leadconnector_post_patch_$(date +%Y%m%d).sql'. Diff this export against the most recent clean backup SQL dump using 'diff]'" to isolate specific records modified during the exploitation window. For 72-hour post-patch monitoring with no SIEM, configure a cron job running every 15 minutes to grep Nginx/Apache logs for POST/PUT to /wp-json/leadconnector/ and email alerts on any HTTP 200 responses: 'grep -E "(POST|PUT).*wp-json/leadconnector.*200" /var/log/nginx/access.log | mail -s "LeadConnector anomaly alert" soc@yourorg.com'.

Evidence: Before restoring from backup, export and preserve the current state of all WordPress database tables associated with LeadConnector (wp_leadconnector_leads, wp_leadconnector_forms, and related wp_options entries) as tampered-state forensic evidence. Document specific lead record IDs, submission timestamps, and field values that differ from the pre-exploitation backup, as these represent the direct output of the attacker's unauthorized REST API write operations and may be needed for breach notification analysis if PII was overwritten or exfiltrated.

Step 5: Post-Incident — This vulnerability exposes a control gap in plugin vetting and REST API exposure management. Implement a WordPress plugin approval process requiring security review before deployment. Add REST API route auditing to your web application scanning program. Review all WordPress plugins for unauthenticated API surface using tools such as WPScan against your environment on a scheduled basis.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned review of the LeadConnector authorization bypass to update plugin governance, REST API exposure controls, and detection capabilities to prevent recurrence of unauthenticated REST API exploitation across the WordPress estate.

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-2 (Flaw Remediation), NIST RA-3 (Risk Assessment), 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: Schedule monthly WPScan runs against all WordPress instances to enumerate registered REST API namespaces and flag unauthenticated endpoints: 'wpscan --url https://yoursite.com --enumerate ap --plugins-detection aggressive'. Write a Sigma rule targeting web logs for unauthenticated POST/PUT to /wp-json/ paths to catch similar plugin authorization bypasses across the entire WordPress fleet. Maintain a plugin inventory spreadsheet tracking name, version, last-reviewed date, and REST API surface for each installed plugin, reviewed quarterly or upon any new CISA KEV addition affecting WordPress plugins.

Evidence: Compile a final incident timeline document mapping attacker source IPs, first and last observed exploitation timestamps from web/WAF logs, specific REST API endpoints called, volume of unauthorized writes, and any data records confirmed tampered — this timeline serves as the evidentiary basis for the lessons-learned review and any required regulatory breach notification determination. Retain all collected log exports, database snapshots, and plugin file captures from prior steps per your organization's incident record retention policy, referencing NIST AU-11 (Audit Record Retention).

Detection Guidance

Search web server access logs and WAF logs for unauthenticated requests targeting LeadConnector REST API endpoints. Key indicators: requests to paths matching /wp-json/leadconnector/* with no Authorization header

and no valid nonce, HTTP methods POST or PUT, HTTP response code 200 or 201 from an external IP. Multiple such requests from a single IP or ASN may indicate active exploitation attempts. Also inspect the WordPress database (wp_options or plugin-specific tables) for unexpected changes to LeadConnector-stored records with timestamps correlating to suspicious request windows. No public IOCs (IPs, hashes, domains) were available in source data at time of generation.

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1565.001** — Stored Data Manipulation

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
- **AC-3** — Access Enforcement

OWASP-TOP10-2021

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

CIS-V8

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

ISO-27001-2022

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1190	Exploit Public-Facing Application	Initial-Access
T1565.001	Stored Data Manipulation	Impact

Sources

Source	URL	Tier
vulncheck_kev	https://nvd.nist.gov/vuln/detail/CVE-2026-1890	T1

Source	URL	Tier
CVE-2026-1890 : The LeadConnector WordPress plugin before 3.0 ...	https://www.cvedetails.com/cve/CVE-2026-1890/	T3
CVE-2026-1890 | Tenable®	https://www.tenable.com/cve/CVE-2026-1890	T3
WordPress Security Bulletin: LeadConnector Plugin Vulnerability ...	https://freshysites.com/security-bulletins/wordpress-security-bulle...	T3
CVE-2026-1890 - Vulnerability Details - OpenCVE	https://app.opencve.io/cve/CVE-2026-1890	T3
CISA KEV	https://www.cisa.gov/known-exploited-vulnerabilities-catalog	T1

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