INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-14 18:51 UTC

CVE-2026-8181: Critical Authentication Bypass in Burst Statistics WordPress Plugin Enables Unauthenticated Admin Takeover

CVE VULNERABILITY | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0181
Type	CVE Vulnerability
CVE ID	CVE-2026-8181
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.0026 (49th percentile)
Affected Products	Burst Statistics WordPress plugin versions 3.4.0 and 3.4.1; patched in 3.4.2 (estimated 115,000 of ~200,000 active installs remain unpatched as of May 14)
Published	2026-05-14T17:07:17
Discovery Source	Rss

Executive Summary

A critical authentication bypass in the Burst Statistics WordPress plugin (versions 3.4.0 and 3.4.1) allows any unauthenticated attacker to create administrator accounts and take full control of affected sites. Approximately 115,000 of the plugin's 200,000 active installs remain unpatched as of May 14, 2026. Exploitation at scale is confirmed, with threat intelligence indicating significant attack volume in the 24 hours following public disclosure. Organizations running WordPress sites with this plugin face complete site compromise, including data theft, defacement, and malware deployment, until the patch is applied.

Technical Analysis

CVE-2026-8181 is a critical authentication bypass (CVSS 9.5) in the Burst Statistics WordPress plugin, introduced in version 3.4.0 (released April 23, 2026) and present in 3.4.1. The vulnerability stems from improper authentication logic in the plugin's REST API endpoint, which can be abused by unauthenticated remote attackers to impersonate administrator-level users and create rogue admin accounts, resulting in full WordPress site takeover without any credentials. Classified under CWE-287 (Improper Authentication), CWE-303 (Incorrect Implementation of Authentication Algorithm), and CWE-807 (Reliance on Untrusted Inputs in a Security Decision). Relevant MITRE ATT&CK techniques include T1190 (Exploit Public-Facing Application), T1078/T1078.001 (Valid Accounts / Default Accounts), T1136.001 (Create Local Account), T1098 (Account

Manipulation), T1556 (Modify Authentication Process), and T1059 (Command and Scripting Interpreter). Active exploitation is confirmed. The patch is available in version 3.4.2. Sources: NVD (nvd.nist.gov, T1), CVE.org, vendor security advisory, GitHub Advisory GHSA-qv3x-rrx4-9pmh.

Action Checklist

1. Step 1: Containment. Identify all WordPress installations in your environment running Burst Statistics versions 3.4.0 or 3.4.1. If immediate patching is not possible, disable the plugin entirely to eliminate the REST API attack surface. Apply WAF rules blocking unauthenticated POST requests to Burst Statistics REST API endpoints as a temporary control.
2. Step 2: Detection. Query WordPress user tables and audit logs for administrator accounts created on or after April 23, 2026 that cannot be attributed to legitimate administrative activity. Review web server and WAF logs for anomalous REST API requests targeting Burst Statistics endpoints (path pattern: /wp-json/burst/*). Look for POST requests to these endpoints originating from untrusted IPs, especially those returning HTTP 200 responses. (See Detection Guidance section for detailed SQL query and log patterns.)
3. Step 3: Eradication. Update the Burst Statistics plugin to version 3.4.2 immediately via the WordPress admin dashboard or WP-CLI. Remove any rogue administrator accounts identified during detection. Rotate credentials for all legitimate administrator accounts on affected sites as a precaution, as attacker persistence via account manipulation (T1098) is a confirmed post-exploitation path.
4. Step 4: Recovery. After patching, verify the installed plugin version is 3.4.2 or later. Re-audit the WordPress user table to confirm no unauthorized admin accounts remain. Review file system integrity for web shells or malicious file uploads deposited during any exploitation window. Monitor WAF and authentication logs for continued exploitation attempts against the patched endpoint for at least 72 hours post-remediation.
5. Step 5: Post-Incident. This incident exposes a gap in change-management controls for third-party WordPress plugins: a vulnerability was introduced in a named version release (3.4.0, April 23) and mass exploitation began before a significant portion of the install base patched. Implement automated plugin version monitoring and alerting. Establish a policy requiring WAF rules or virtual patching for critical WordPress plugin CVEs within 24 hours of disclosure. Review whether WordPress REST API endpoints are exposed without authentication controls at the network or application layer.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to senior IR leadership, legal counsel, and executive stakeholders if any rogue administrator account is confirmed to have been used to access, export, or modify user data (wp_users, WooCommerce order tables, contact form submissions), as this may trigger breach notification obligations under GDPR Article 33, CCPA, or applicable state breach notification laws; also escalate immediately if web shells or unauthorized plugin installations are discovered, indicating the attacker progressed beyond account creation to full site compromise.

Recovery Notes	After patching to Burst Statistics 3.4.2 and removing rogue accounts, maintain heightened monitoring of /wp-json/burst/ POST requests and WordPress authentication logs for a minimum of 72 hours given the active campaign's rate of 7,400+ attacks per day — any HTTP 200 responses to these endpoints post-patch indicate a possible exploit bypass or incomplete remediation. Verify file system integrity in wp-content/uploads/ and wp-content/plugins/ for web shells or unauthorized plugin installations that may have been staged during any confirmed exploitation window before the patch was applied. Confirm outbound network connections from the web server during the exploitation window via web server error logs and, if available, netflow data, as attackers with admin access may have established C2 channels or exfiltrated data independent of the plugin vulnerability itself.
Forensic Artifacts	wp_users and wp_usermeta database tables: cross-reference user_registered timestamps against April 23, 2026 (version 3.4.0 release date) and filter wp_capabilities for 'administrator' role — rogue accounts created via CVE-2026-8181 will appear in this window with no corresponding legitimate change-management record | Web server access logs (Apache access.log / Nginx access.log): filter on 'POST' method and URI path '/wp-json/burst/' with HTTP 200 response codes — successful exploit requests targeting the Burst Statistics authentication bypass REST API endpoint will appear as small POST requests (~200-500 bytes) returning 200 OK from IPs with no prior authenticated session history | WAF event logs (Cloudflare, Sucuri, or ModSecurity audit log at /var/log/modsec_audit.log): review blocked and allowed events for the /wp-json/burst/ path from April 23 onward — the 7,400+ attacks-per-day campaign rate will produce a distinct spike in WAF telemetry attributable to automated exploitation tooling with consistent User-Agent strings or request structures across source IPs | WordPress wp-content/uploads/ directory: enumerate all .php, .phtml, .asp, and .shtml files created after April 23, 2026 using 'find wp-content/uploads -name '*.php' -newermt 2026-04-23' — attackers who successfully created admin accounts may have used the media upload or theme editor functionality to deploy web shells in this directory, which is typically not scanned by default file integrity monitors | wp_options table: query for rows with option_name IN ('active_plugins','siteurl','admin_email','blogname') and compare against a known-good baseline — an attacker with admin access gained via CVE-2026-8181 may have modified site settings, activated malicious plugins, or redirected the site URL as part of post-exploitation activity following the T1098 account manipulation phase

Per-Action IR Details

Step 1: Containment — Identify all WordPress installations in your environment running Burst Statistics versions 3.4.0 or 3.4.1. If immediate patching is not possible, disable the plugin entirely to eliminate the REST API attack surface. Apply WAF rules blocking unauthenticated POST requests to Burst Statistics REST API endpoints as a temporary control.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: prioritize stopping ongoing damage by isolating the vulnerable attack surface before full eradication is possible

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality) — restrict REST API exposure to authenticated requests only, CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: Run 'wp plugin list --fields=name,version,status --format=csv' via WP-CLI across all managed WordPress installs to enumerate Burst Statistics version 3.4.0 or 3.4.1. For teams managing multiple sites without a central dashboard, use a bash one-liner: 'find /var/www -name 'plugin.php' -path '*/burst-statistics/*' | xargs grep -l 'Version: 3.4''. Disable with 'wp plugin deactivate burst-statistics'. Add a Nginx or Apache rewrite rule to return HTTP 403 for any request matching '^/wp-json/burst/' from unauthenticated sources as a zero-cost WAF substitute.

Evidence: Before disabling the plugin, capture: (1) the current plugin file contents from wp-content/plugins/burst-statistics/ to establish which version is installed and whether files have been tampered with since installation; (2) a full export of the wp_users and wp_usermeta tables timestamped at capture time to establish the pre-containment admin account baseline; (3) web server access logs (Apache: /var/log/apache2/access.log; Nginx: /var/log/nginx/access.log) covering April 23, 2026 through containment date, filtered on POST requests to /wp-json/burst/; (4) WAF alert logs if a cloud WAF (Cloudflare, Sucuri) is in use, exported before any rule changes flush cached events.

Step 2: Detection — Query WordPress user tables and audit logs for administrator accounts created on or after April 23, 2026 that cannot be attributed to legitimate administrative activity. Review web server and WAF logs for anomalous REST API requests targeting Burst Statistics endpoints (path pattern: /wp-json/burst/*). Look for POST requests to these endpoints originating from untrusted IPs, especially those returning HTTP 200 responses. Check for newly created wp_users entries with admin role assignment.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate log sources to identify the scope of exploitation; April 23, 2026 is the confirmed introduction date of the vulnerable code in version 3.4.0 and serves as the earliest possible exploitation timestamp

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

Compensating: Query the WordPress database directly: 'SELECT u.ID, u.user_login, u.user_registered, u.user_email FROM wp_users u INNER JOIN wp_usermeta m ON u.ID = m.user_id WHERE m.meta_key = 'wp_capabilities' AND m.meta_value LIKE '%administrator%' AND u.user_registered >= '2026-04-23';'. For log analysis without a SIEM, use 'grep -E 'POST.*wp-json/burst' /var/log/nginx/access.log | awk '{print $1, $7, $9}' | sort | uniq -c | sort -rn' to surface high-frequency source IPs with HTTP 200 responses. Install the free WP Activity Log plugin (version 4.x) retroactively if audit logging was not previously enabled — it can reconstruct some events from existing WordPress database records.

Evidence: Capture before any account deletion or log rotation: (1) full wp_users table dump including user_registered timestamps, cross-referenced against wp_usermeta for 'wp_capabilities' = 'administrator'; (2) web server access logs filtered on 'POST /wp-json/burst/' with HTTP response code 200, preserving source IP, timestamp, User-Agent, and request body size — attacker POST bodies to the Burst Statistics registration endpoint will be notably small and structurally uniform across the attack campaign; (3) WAF logs showing blocked vs. allowed requests to /wp-json/burst/ to differentiate probing from successful exploitation; (4) WordPress debug.log (wp-content/debug.log if WP_DEBUG_LOG is enabled) for PHP errors triggered by malformed exploit payloads from failed attempts.

Step 3: Eradication — Update the Burst Statistics plugin to version 3.4.2 immediately via the WordPress admin dashboard or WP-CLI ('wp plugin update burst-statistics'). Remove any rogue administrator accounts identified during detection. Rotate credentials for all legitimate administrator accounts on affected sites as a precaution, as attacker persistence via account manipulation (T1098) is a confirmed post-exploitation path.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the vulnerability (patch to 3.4.2) and all attacker-established persistence mechanisms (rogue admin accounts) before restoration; MITRE ATT&CK T1098 (Account Manipulation) is the confirmed post-exploitation persistence technique observed in this campaign

Controls: NIST SI-2 (Flaw Remediation), NIST IR-4 (Incident Handling), NIST AC-2 (Account Management) — revoke unauthorized accounts created via CVE-2026-8181, NIST IA-5 (Authenticator Management) — rotate compromised credentials, CIS 7.4 (Perform Automated Application Patch Management), CIS 5.3 (Disable Dormant Accounts), CIS 5.1 (Establish and Maintain an Inventory of Accounts)

Compensating: Patch via WP-CLI: 'wp plugin update burst-statistics --version=3.4.2' then verify with 'wp plugin get burst-statistics --field=version'. Remove rogue accounts: 'wp user delete --reassign=' for each account identified in Step 2. Rotate all legitimate admin passwords: 'wp user update --user_pass=$(openssl rand -base64 24)' and force logout of all active sessions with 'wp user session destroy --all'. If the site uses application passwords (WordPress 5.6+), revoke all application passwords for admin accounts: accessible via wp-admin → Users → Application Passwords.

Evidence: Before deleting rogue accounts, preserve: (1) full account profile data from wp_users and wp_usermeta for each rogue administrator (user_login, user_email, user_registered, capabilities, last login metadata) to support threat actor attribution and potential legal referral; (2) any posts, pages, or uploaded files created by the rogue account user IDs — run 'SELECT ID, post_title, post_date, post_status, post_type FROM wp_posts WHERE post_author = ';' to identify content the attacker may have staged; (3) wp_options table entries modified on or after April 23, 2026, particularly 'siteurl', 'admin_email', 'blogdescription', and active plugin list, as attackers with admin access may have altered site configuration or installed malicious plugins.

Step 4: Recovery — After patching, verify the installed plugin version is 3.4.2 or later. Re-audit the WordPress user table to confirm no unauthorized admin accounts remain. Review file system integrity for web shells or malicious file uploads deposited during any exploitation window. Monitor WAF and authentication logs for continued exploitation attempts against the patched endpoint for at least 72 hours post-remediation.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore the system to verified clean state and confirm threat removal before resuming normal operations; 72-hour monitoring window aligns with the campaign's observed exploitation velocity of 7,400+ attacks per 24-hour period

Controls: NIST SI-7 (Software, Firmware, and Information Integrity), NIST IR-4 (Incident Handling), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST CM-3 (Configuration Change Control) — verify only authorized plugin version is running, CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Verify patch: 'wp plugin get burst-statistics --field=version' must return '3.4.2'. File integrity check: run 'wp core verify-checksums' for WordPress core, then compare plugin file hashes against the official 3.4.2 release from wordpress.org using 'md5sum wp-content/plugins/burst-statistics/*.php' and cross-reference against the published package. For web shell detection without a commercial scanner, run: 'find wp-content/uploads -name '*.php' -newer wp-content/plugins/burst-statistics/burst-statistics.php' to identify PHP files written to the uploads directory (a common web shell staging path) during the exploitation window. Use the free plugin Wordfence (free tier) or WPScan (open source) to scan for known malicious file signatures.

Evidence: Capture post-recovery verification artifacts for the incident record: (1) 'wp plugin list' output showing burst-statistics at version 3.4.2 with status 'active', timestamped; (2) final wp_users query output confirming zero unauthorized administrator accounts remain; (3) file system listing of wp-content/uploads/ filtered on files created between April 23, 2026 and the patch date, noting any .php, .phtml, or .shtml files which would indicate web shell staging by an attacker who leveraged the admin takeover to upload malicious content; (4) WAF/access log baseline for /wp-json/burst/ POST requests post-patch to establish normal traffic pattern for the 72-hour monitoring comparison.

Step 5: Post-Incident — This incident exposes a gap in change-management controls for third-party WordPress plugins: a vulnerability was introduced in a named version release (3.4.0, April 23) and mass exploitation began before a significant portion of the install base patched. Implement automated plugin version monitoring and alerting. Establish a policy requiring WAF rules or virtual patching for critical WordPress plugin CVEs within 24 hours of disclosure. Review whether WordPress REST API endpoints are exposed without authentication controls at the network or application layer.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned to address the specific control gap — absence of automated plugin version monitoring allowed 115,000 installs to remain on vulnerable versions despite a patch being available; update IR plan and detection capabilities accordingly

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan) — update plan to include WordPress plugin CVE response procedures, NIST SI-5 (Security Alerts, Advisories, and Directives) — establish formal intake for WordPress plugin CVE disclosures, NIST SI-2 (Flaw Remediation) — formalize SLA for critical plugin CVE remediation, NIST CM-7 (Least Functionality) — evaluate whether /wp-json/ should be restricted at the network layer by default, CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Implement automated plugin version monitoring using WPScan's free API (250 requests/day) in a cron job: 'wpscan --url https://yoursite.com --enumerate p --plugins-detection aggressive --api-token ' scheduled daily, with output piped to a script that alerts on any plugin at a known-vulnerable version. Subscribe to the Wordfence Intelligence feed (free) and the WordPress Plugin Security Team's disclosure channel to receive CVE notifications before mass exploitation begins. For REST API hardening without enterprise tooling, add 'add_filter('rest_authentication_errors', function($result){ if(!is_user_logged_in()){ return new WP_Error('rest_not_logged_in','Authentication required.',array('status'=>401));} return $result;});' to functions.php to require authentication for all REST API calls except explicitly whitelisted routes.

Evidence: Compile the post-incident record with: (1) timeline of events mapping plugin version 3.4.0 release (April 23, 2026) → CVE disclosure → first exploitation evidence in your logs → containment action, to quantify the exposure window specific to your environment; (2) the final count and profiles of any unauthorized administrator accounts created, correlated against source IPs from web server logs, to support threat intelligence sharing with Wordfence, CISA, or sector ISACs; (3) documentation of which sites in your inventory were on 3.4.0/3.4.1 versus already on 3.4.2 or had plugin disabled, to measure the asset inventory and patch visibility gap that allowed this exposure.

Detection Guidance

Primary detection target: unauthorized administrator account creation on WordPress sites running Burst Statistics 3.4.0 or 3.4.1. Query the WordPress database directly: 'SELECT user_login, user_registered, meta_value FROM wp_users JOIN wp_usermeta ON wp_users.ID = wp_usermeta.user_id WHERE meta_key = "wp_capabilities" AND meta_value LIKE "%administrator%" AND user_registered >= "2026-04-23"'. Compare results against your known admin roster. In web server or WAF logs, look for: unauthenticated POST requests to paths matching '/wp-json/burst/' or '/wp-json/burst-statistics/', HTTP 200 responses to those requests from unknown source IPs, and high-frequency requests to these endpoints from single IPs (indicative of automated scanning). Behavioral indicators post-exploitation include: new admin-level logins from unfamiliar geographic locations or IPs, plugin or theme file modifications via the WordPress admin interface, and new PHP files in wp-content directories not corresponding to known deployments. As of the data collection date, no IOCs (malicious IPs, file hashes, or domains) were published by NVD or vendor advisory. Check CISA KEV, Shodan, or threat intelligence feeds for emerging IOC data.

Framework Mappings

MITRE-ATTACK

- **T1589.002** — Email Addresses
- **T1078** — Valid Accounts
- **T1556** — Modify Authentication Process
- **T1098** — Account Manipulation
- **T1059** — Command and Scripting Interpreter
- **T1078.001** — Default Accounts
- **T1136.001** — Local Account
- **T1190** — Exploit Public-Facing Application

NIST-800-53R5

- **AC-2** — Account Management
- **AC-6** — Least Privilege

- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **IA-8** — Identification and Authentication (Non-Organizational Users)
- **IR-5** — Incident Monitoring

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication

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
T1589.002	Email Addresses	Reconnaissance
T1078	Valid Accounts	Defense-Evasion

Technique ID	Technique Name	Tactic
T1556	Modify Authentication Process	Credential-Access
T1098	Account Manipulation	Persistence
T1059	Command and Scripting Interpreter	Execution
T1078.001	Default Accounts	Defense-Evasion
T1136.001	Local Account	Persistence
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/hackers-exploit-auth...	T3
CVE-2026-8181 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-8181	T1
CVE-2026-8181 — Remote Code Execution in Burst Statistics | dbugs	https://dbugs.ptsecurity.com/vulnerability/PT-2026-40880	T3
CVE-2026-8181 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-8181	T3
The Burst Statistics – Privacy-Friendly WordPress... · CVE-2026-8181	https://github.com/advisories/GHSA-qv3x-rrx4-9pmh	T3

Generated 2026-05-14 18:51 UTC by TJS Security Command Center