INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-02 06:46 UTC

CVE-2026-42377: Missing Authorization vulnerability in Brainstorm Force SureForms Pro allows Exploiting Incorrectly ...

CVE VULNERABILITY | HIGH | CVSS 7.3

SCC Item ID	SCC-CVE-2026-0111
Type	CVE Vulnerability
CVE ID	CVE-2026-42377
Severity	HIGH
CVSS Base Score	7.3
EPSS Score	0.0001 (2th percentile)
Affected Products	Brainstorm Force SureForms Pro <= 2.8.0
Published	2026-04-29T08:16:18.337
Discovery Source	Nvd

Executive Summary

A high-severity authorization flaw (CVE-2026-42377, CVSS 7.3) exists in Brainstorm Force SureForms Pro, a WordPress form-building plugin, affecting all versions through 2.8.0. Attackers can exploit misconfigured access controls to reach restricted functionality or data without proper permission checks. Organizations running this plugin on public-facing WordPress sites should treat this as a priority patching item.

Technical Analysis

CVE-2026-42377 is a Missing Authorization vulnerability (CWE-862) in Brainstorm Force SureForms Pro <= 2.8.0. The flaw stems from incorrectly configured access control security levels within the WordPress plugin, allowing requests to reach privileged endpoints or data without proper capability checks. MITRE ATT&CK mappings include T1190 (Exploit Public-Facing Application) and T1078 (Valid Accounts); the latter applies because the flaw may be exploited using low-privilege or subscriber-level credentials, or potentially without authentication depending on WordPress configuration. CVSS base score is 7.3 (High); no CVSS vector string is published in available data. EPSS score is 0.00012 (1.6th percentile), indicating low current exploitation activity in the wild. The vulnerability is not listed on the CISA Known Exploited Vulnerabilities catalog as of this report. No vendor CVSS score is available. Remediation requires upgrading beyond version 2.8.0; consult the WordPress plugin repository or Brainstorm Force's official advisory for the patched release version.

Action Checklist

1. Step 1: Containment - Identify all WordPress instances running SureForms Pro <= 2.8.0. If immediate patching is not possible, consider restricting access to the affected plugin's endpoints via WAF rules targeting unauthorized form or admin API requests, or temporarily deactivating the plugin on externally accessible sites.
2. Step 2: Detection - Review WordPress and web server access logs for unexpected requests to SureForms Pro plugin endpoints (wp-admin, wp-json, or plugin-specific REST API routes) from unauthenticated or low-privilege sources. Look for HTTP 200 responses to requests that should require elevated permissions. No public IOCs or exploit signatures are available at this time.
3. Step 3: Eradication - Update SureForms Pro to the latest version released after 2.8.0 via the WordPress admin dashboard or wp-cli. Verify the update through the plugin's changelog for explicit CVE-2026-42377 remediation confirmation. Cross-reference against the Brainstorm Force official advisory once published.
4. Step 4: Recovery - After patching, confirm plugin version in WordPress admin and review access logs for any anomalous activity that occurred before remediation. Validate that form data and plugin-controlled content remain intact and have not been altered. Re-enable the plugin on any sites where it was temporarily deactivated.
5. Step 5: Post-Incident - Assess whether your WordPress deployment process includes automated plugin version monitoring and alerting. This vulnerability exposes a control gap in plugin lifecycle management; consider integrating a WordPress security scanner (such as WPScan or Wordfence) into your routine scanning cadence to detect missing-authorization class issues earlier.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to incident commander and legal/privacy counsel immediately if web server access logs confirm HTTP 200 responses to unauthenticated SureForms Pro REST API requests prior to patching, as successful exploitation of this missing-authorization flaw may have exposed WordPress form submission data (potentially including PII) to unauthorized access, triggering breach notification obligations under GDPR, CCPA, or HIPAA depending on data classification and jurisdiction.
Recovery Notes	After applying the SureForms Pro patch, monitor web server access logs for continued probing of `/wp-json/sureforms/` and `/wp-admin/admin-ajax.php` SureForms routes for a minimum of 72 hours, as threat actors who successfully enumerated the endpoint pre-patch may retry post-patch to confirm remediation or pivot to other WordPress plugin attack surface. Validate the integrity of all form submission records in the WordPress database (wp_posts where post_type relates to SureForms entries) against pre-incident backups to confirm no unauthorized data exfiltration or tampering occurred during the exposure window. If the site accepts sensitive form submissions (contact forms, registration data, payment-adjacent workflows), notify your privacy officer to assess whether the exposure window constitutes a reportable data event before closing the incident.

Forensic Artifacts	Web server access logs (/var/log/nginx/access.log or /var/log/apache2/access.log): Filter for HTTP 200 responses to /wp-json/sureforms/ and /wp-admin/admin-ajax.php?action=sureforms_* from IPs with no corresponding authenticated WordPress session — the missing-authorization flaw means successful exploitation produces a 200 rather than a 401/403, making response code the primary discrimination signal. | WordPress database wp_posts and wp_postmeta tables: Query for SureForms form entries (post_type = 'sureforms_form') created or modified by user_id = 0 (unauthenticated) or subscriber-level roles during the vulnerability exposure window, which would indicate an attacker leveraged the missing permission check to interact with restricted form or data endpoints. | WordPress wp_options table rows prefixed 'sureforms': Capture pre- and post-incident snapshots to detect unauthorized modification of plugin configuration, such as attacker-altered notification endpoints, webhook destinations, or form access control settings that could indicate persistence or data exfiltration setup via the misconfigured plugin. | PHP error log and WordPress debug log (WP_DEBUG_LOG at /wp-content/debug.log): Examine for PHP warnings or fatal errors originating from SureForms Pro files during the exposure window — authorization bypass attempts against improperly guarded REST callbacks frequently generate PHP notices that reveal the specific function and capability check that was circumvented. | WordPress user meta table (wp_usermeta, meta_key = 'wp_capabilities'): Review for any unexpected privilege escalations — particularly subscriber or unauthenticated users gaining elevated capabilities — timestamped within the CVE-2026-42377 exposure window, as missing-authorization vulnerabilities in form plugins have been chained with user role manipulation in prior WordPress plugin exploit chains (e.g., similar patterns in CVE-2024-11205, Woffice theme).

Per-Action IR Details

Step 1: Containment — Identify all WordPress instances running SureForms Pro <= 2.8.0. If immediate patching is not possible, consider restricting access to the affected plugin's endpoints via WAF rules targeting unauthorized form or admin API requests, or temporarily deactivating the plugin on externally accessible sites.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST CM-7 (Least Functionality), 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 --status=active --format=csv | grep sureforms` via WP-CLI across all managed WordPress instances to enumerate exposed installations. For WAF-less environments, use Apache/Nginx rewrite rules to block unauthenticated requests to `/wp-json/sureforms/` and `/wp-admin/admin-ajax.php?action=sureforms_*` endpoints: `location ~* /wp-json/sureforms/ { deny all; }`. Alternatively, use the free Wordfence plugin (firewall tier) to create a custom blocking rule targeting SureForms REST routes from non-authenticated sessions. Document deactivation timestamps per site for the incident timeline.

Evidence: Before deactivating or blocking, capture: (1) current Apache/Nginx access logs (`/var/log/apache2/access.log` or `/var/log/nginx/access.log`) showing requests to `/wp-json/sureforms/`, `/wp-admin/admin-ajax.php`, and plugin-specific REST API routes — preserve with `cp` and hash with `sha256sum`; (2) WordPress database snapshot (`wp db export pre-containment-$(date +%F).sql`) to baseline form data and plugin option values in `wp_options` table (rows with `option_name LIKE 'sureforms%'`); (3) PHP error log (`/var/log/php*.log` or `WP_DEBUG_LOG` output) for anomalous authorization bypass attempts; (4) Current plugin file state via `wp plugin verify-checksums brainstorm-force-sureforms-pro` to detect any file tampering prior to remediation.

Step 2: Detection — Review WordPress and web server access logs for unexpected requests to SureForms Pro plugin endpoints (wp-admin, wp-json, or plugin-specific REST API routes) from unauthenticated or low-privilege sources. Look for HTTP 200 responses to requests that should require elevated permissions. No

public IOCs or exploit signatures are available at this time.

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-3 (Content of Audit Records), CIS 8.2 (Collect Audit Logs)

Compensating: Execute the following grep against web server access logs to surface HTTP 200 responses to SureForms REST endpoints from unauthenticated sources: `grep -E '(wp-json/sureforms|admin-ajax.php\?action=sureforms)' /var/log/nginx/access.log | grep ' 200 '`. Cross-reference source IPs against authentication logs (`/var/log/auth.log` or WordPress `wp_usermeta` last-login entries) to confirm whether those IPs correspond to authenticated sessions. Use GoAccess (free, CLI-based) to parse and visualize access log frequency spikes: `goaccess /var/log/nginx/access.log --log-format=COMBINED`. Deploy the free Sigma rule for WordPress REST API abuse (community rule `web.wordpress_rest_api_suspicious_request`) via a log-shipping script to a local Wazuh or Graylog instance if available.

Evidence: Preserve the following before any log rotation occurs: (1) Full web server access logs for the 90 days prior to discovery, filtered for requests to `/wp-json/sureforms/v*/` and `/wp-admin/admin-ajax.php` — HTTP method, source IP, user-agent, response code, and response body size are all relevant to distinguishing reconnaissance from successful exploitation; (2) WordPress `wp_posts` and `wp_postmeta` database tables for any form entries (post_type = `sureforms_form`) created or modified by unexpected user IDs, particularly `user_id = 0` (unauthenticated) or low-privilege subscriber/contributor roles; (3) PHP session files in `/tmp` or the configured session directory for anomalous session tokens concurrent with suspicious access log entries; (4) WordPress user activity via `wp_usermeta` for unexpected privilege escalations (meta_key = `wp_capabilities`) timestamped near exploit window.

Step 3: Eradication — Update SureForms Pro to the latest version released after 2.8.0 via the WordPress admin dashboard or wp-cli. Verify the update through the plugin's changelog for explicit CVE-2026-42377 remediation confirmation. Cross-reference against the Brainstorm Force official advisory once published.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), 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: Apply the patch via WP-CLI with integrity verification: `wp plugin update brainstorm-force-sureforms-pro --version=` followed immediately by `wp plugin verify-checksums brainstorm-force-sureforms-pro` to confirm installed files match the WordPress.org repository checksums. If the plugin is distributed outside WordPress.org (premium channel), download the patched ZIP from the Brainstorm Force account portal, verify the SHA-256 hash against the vendor-published value, and install via `wp plugin install /path/to/sureforms-pro-patched.zip --force`. After update, audit plugin PHP files for the missing authorization check by grepping for the affected REST route registration: `grep -r 'register_rest_route\|permission_callback' /wp-content/plugins/sureforms-pro/ | grep -i 'false\|__return_true'` — the patched version should replace permissive callbacks with proper capability checks.

Evidence: Before applying the patch, preserve: (1) Full copy of the vulnerable plugin directory (`/wp-content/plugins/sureforms-pro/` or equivalent) archived and hashed — this is the forensic baseline for determining what authorization logic was absent and what an attacker could have accessed; (2) Output of `wp plugin get brainstorm-force-sureforms-pro --format=json` capturing installed version, auto-update status, and file modification timestamps; (3) Git diff or file comparison if the WordPress installation is version-controlled, to detect any attacker-introduced modifications to plugin files prior to patching; (4) Database export of `wp_options` rows prefixed with `sureforms` to document plugin configuration state at time of eradication.

Step 4: Recovery — After patching, confirm plugin version in WordPress admin and verify access logs for any anomalous activity that occurred before remediation. Validate that form data and plugin-controlled content remain intact and have not been altered. Re-enable the plugin on any sites where it was temporarily deactivated.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Confirm patched version with `wp plugin get brainstorm-force-sureforms-pro --field=version` and cross-reference against Brainstorm Force changelog for explicit CVE-2026-42377 mention. Validate data integrity of SureForms form submissions by querying the WordPress database: `wp db query "SELECT ID, post_author, post_date, post_modified, post_status FROM wp_posts WHERE post_type='sureforms_form' ORDER BY post_modified DESC LIMIT 50;"` — flag any entries modified by user_id 0 or within the exploit window. Re-enable the plugin only after verifying WAF rules or access controls remain in place, and monitor the first 72 hours of re-enabled operation by tailing access logs: `tail -f /var/log/nginx/access.log | grep -E 'wp-json/sureforms|sureforms_'`.

Evidence: Preserve for post-recovery validation: (1) WordPress admin audit trail (if Wordfence or WP Activity Log plugin is installed) showing plugin re-activation events and the authenticated user performing recovery actions; (2) Post-patch access log snapshot covering the first 24 hours of re-enabled plugin operation, filtered for SureForms endpoints, to establish a clean baseline and detect any resumed exploitation attempts; (3) Database comparison of `wp_posts` (post_type = `sureforms_form`) and `wp_postmeta` between pre-containment export and post-recovery state to identify any unauthorized form data creation, deletion, or modification; (4) Screenshot or export of WordPress admin Dashboard > Plugins page showing confirmed patched version number as a timestamped record for the incident log.

Step 5: Post-Incident — Assess whether your WordPress deployment process includes automated plugin version monitoring and alerting. This vulnerability exposes a control gap in plugin lifecycle management; consider integrating a WordPress security scanner (such as WPScan or Wordfence) into your routine scanning cadence to detect missing-authorization class issues earlier.

NIST Phase: Post Incident

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

Controls: NIST IR-4 (Incident Handling), NIST SI-5 (Security Alerts, Advisories, and Directives), 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 WPScan (free tier, CLI) as a weekly cron job across all WordPress instances: `wpscan --url https://yoursite.com --enumerate p --plugins-detection aggressive --api-token ` — the free API tier provides CVE lookup for installed plugins including missing-authorization class findings. For plugin version drift monitoring without a paid tool, use a Bash script querying WP-CLI across all managed sites: `for site in $(wp site list --field=url); do wp --url=$site plugin list --status=active --format=csv; done` piped to a diff against a known-good baseline. Subscribe to the Brainstorm Force security mailing list and WPScan Vulnerability Database RSS feed (`https://wpscan.com/feed/`) to receive future advisories for SureForms Pro and other Brainstorm Force products without relying on vendor-push updates.

Evidence: Document and retain for lessons learned: (1) Timeline reconstruction from web server access logs showing first observed request to SureForms REST endpoints, earliest possible exploitation window, and detection-to-containment gap — this quantifies dwell time specific to this CVE-2026-42377 incident; (2) Inventory of all WordPress plugin versions across the environment at time of discovery (output of `wp plugin list` per site), retained to document the scope of the plugin lifecycle gap exposed by this vulnerability; (3) WAF rule logs (if deployed) showing whether any SureForms endpoint abuse was blocked prior to manual detection — absence of WAF coverage is itself a finding for the lessons-learned report; (4) Wordfence or WPScan scan report from the post-patch scan run, confirming CVE-2026-42377 is no longer flagged and documenting any additional missing-authorization findings in co-installed plugins.

Detection Guidance

No public exploit code or specific IOCs are available for CVE-2026-42377 at this time. Detection should focus on behavioral indicators in web server and WordPress logs: look for unauthenticated or low-privilege requests

returning HTTP 200 to plugin REST API routes or admin-only endpoints associated with SureForms Pro (typically under /wp-json/ or /wp-admin/admin-ajax.php with SureForms action parameters). Flag any access to form submission, data export, or configuration endpoints from sessions without expected WordPress capability levels. This requires WordPress-aware logging (e.g., Wordfence, WPScan, or custom log parsing) to be effective; if not in place, prioritize HTTP 200 responses to /wp-json/ or /wp-admin/admin-ajax.php endpoints from unexpected source IPs or user agents. Web application firewall logs should be reviewed for rule triggers on plugin-path requests. If Wordfence or a similar WordPress security plugin is deployed, check its blocked request log for attempts against SureForms endpoints. EPSS score (0.00012) indicates no significant exploitation activity detected in threat intelligence feeds as of the configuration date.

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1078** — Valid Accounts

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-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **AC-3** — Access Enforcement

OWASP-TOP10-2021

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

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.3** — Require MFA for Externally-Exposed Applications

HIPAA-SECURITY

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

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC6.3** — Authorizes, modifies, or removes access

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
T1078	Valid Accounts	Defense-Evasion

Sources

Source	URL	Tier
nvd	https://nvd.nist.gov/vuln/detail/CVE-2026-42377	T1
CVE-2026-42377	https://www.tenable.com/cve/CVE-2026-42377	T3
CVE-2026-42377 - Exploits & Severity	https://feedly.com/cve/CVE-2026-42377	T3
CVE Explorer – Vulnerability Database | CD - Cyber Defence	https://www.cyber-defence.io/tools/cve/CVE-2026-42377	T3
Missing Authorization vulnerability in Brainstorm Force...	https://github.com/advisories/GHSA-mv53-6q27-4m6w	T3

Generated 2026-05-02 06:46 UTC by TJS Security Command Center