INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-10 06:15 UTC

Smart Slider 3 Pro Update Channel Compromised: Trojanized v3.5.1.35 Delivers Multi-Layer Backdoor to 900K+ Sites

THREAT CAMPAIGN | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CAM-2026-0162
Type	Threat Campaign
Severity	CRITICAL
CVSS Base Score	9.5
Affected Products	Smart Slider 3 Pro v3.5.1.35 (WordPress and Joomla plugins)
Published	2026-04-09T12:15:26
Discovery Source	Rss

Executive Summary

On April 7, 2026, attackers hijacked the update delivery infrastructure for Smart Slider 3 Pro and pushed a malicious build (v3.5.1.35) to over 900,000 WordPress sites and an unknown number of Joomla installations. The trojanized update installs a backdoor that gives attackers unauthenticated remote access, the ability to create hidden administrator accounts, and persistence mechanisms that survive password resets and standard remediation. Any site that auto-updated during the compromise window must be treated as fully compromised until forensically cleared.

Technical Analysis

Affected product: Smart Slider 3 Pro v3.5.1.35 (WordPress and Joomla). Attack vector: supply chain compromise of the vendor update channel (MITRE T1195.002). The trojanized build embeds a multi-capability implant with the following confirmed behaviors: unauthenticated remote code execution via PHP eval-equivalent execution (T1059 parent; no strict MITRE equivalent for server-side PHP eval); OS command injection (CWE-78, T1059.004); unrestricted file upload enabling web shell deployment (CWE-434, T1505.003); covert local administrator account creation (CWE-798 hard-coded credentials, T1136.001); event-triggered persistence via mu-plugin autoload (T1546); and authentication process modification (T1556). Integrity-check bypass (CWE-494) allowed the malicious build to pass standard update verification. Embedded malicious code classification: CWE-506. As of the initial advisory date (April 9, 2026), no CVE identifier has been assigned. CVSS assessment reflects campaign-level impact (9.5 criticality). Primary tracking sources: Wordfence advisory and Smart Slider vendor advisory (HelpScout). No threat actor attribution has been disclosed. Compromise

window: April 7, 2026, through vendor patch release date.

Action Checklist

1. Step 1: Containment, Immediately disable or take offline any WordPress or Joomla site running Smart Slider 3 Pro v3.5.1.35. Block outbound connections from affected web servers at the firewall/WAF layer. Suspend auto-update functionality for all WordPress/Joomla plugins across your estate until this incident is closed (temporary containment measure). Sites that auto-updated during April 7 through the vendor patch release should be isolated and treated as compromised.
2. Step 2: Detection, Audit all WordPress and Joomla installations for Smart Slider 3 Pro version 3.5.1.35 using plugin inventory tools or wp-cli ('wp plugin list --all'). Review wp-content/mu-plugins/ for unrecognized files (T1546 persistence via autoload). Search web server access logs for unauthenticated POST requests to plugin directories. Audit WordPress user tables for unknown administrator accounts (T1136.001). Check file system for recently created .php files in plugin, upload, and mu-plugin directories. Review authentication logs for credential use tied to hard-coded or unknown accounts (CWE-798, T1552).
3. Step 3: Eradication, Update Smart Slider 3 Pro to the vendor-released patched version via the vendor's official channel (confirm version number against the Smart Slider HelpScout advisory before applying). Do not use the auto-update mechanism until vendor confirms update channel integrity is restored. Remove all files associated with the malicious build. Delete any administrator accounts not present prior to April 7. Remove unrecognized files from mu-plugins and upload directories. Restore web root from a known-clean backup predating April 7 if forensic review confirms compromise.
4. Step 4: Recovery, After applying the clean build, re-audit user accounts, mu-plugins directory, and file system for residual artifacts. Reset all WordPress/Joomla administrator credentials and application secrets (salts, API keys). Re-enable the site only after file integrity verification against a pre-compromise baseline. Monitor web application logs and authentication events for 30 days post-remediation for signs of re-entry via persisted access (T1546, T1098). Confirm WAF rules are updated to detect web shell access patterns.
5. Step 5: Post-Incident, This incident exposes a control gap in plugin auto-update trust. Implement a policy requiring integrity verification and staging-environment testing before auto-updates apply to production. Establish a plugin allowlist with version pinning for business-critical WordPress/Joomla plugins. Add plugin directory file integrity monitoring (FIM) to your SIEM detection coverage. Review your software supply chain risk assessment to include third-party plugin update channels as an attack surface.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO, legal counsel, and breach notification counsel immediately if forensic review of `wp_users`, database binary logs, or web server access logs confirms that attacker-created administrator accounts accessed user PII, payment data, or session tokens — triggering potential GDPR Article 33, CCPA, or PCI DSS Requirement 12.10.4 breach notification obligations — or if the organization lacks the internal capability to perform forensic artifact preservation before eradication.

Recovery Notes	Before re-enabling any affected site, validate the restored file tree against a pre-April 7 backup hash manifest and confirm the Smart Slider 3 Pro plugin version matches the vendor-patched build per the HelpScout advisory — do not rely solely on the WordPress admin dashboard version display, which the backdoor could have spoofed. Monitor `wp-content/mu-plugins/`, `wp-content/uploads/`, and the `wp_users` table continuously for 30 days post-recovery, as the multi-layer persistence mechanism (T1546, T1098) described in the advisory may have planted secondary implants in upload directories or serialized PHP objects within `wp_options` that survive a plugin reinstall. Treat any new PHP file appearing in `uploads/` or any new administrator account registered after the recovery timestamp as an active reinfection indicator requiring immediate re-isolation.
Forensic Artifacts	wp-content/mu-plugins/ directory — the trojanized v3.5.1.35 build uses must-use plugin autoload (T1546) to persist a backdoor loader that survives plugin deactivation; any .php file in this directory with a creation or modification timestamp on or after April 7, 2026 UTC is a primary artifact of this campaign | WordPress wp_users and wp_usermeta database tables — the backdoor creates hidden administrator accounts (T1136.001) that may have `user_status` set to suppress visibility in standard admin UI queries; export and diff against a pre-April 7 backup to identify injected accounts and their associated capability metadata in wp_usermeta | Web server access logs (Apache access.log / NGINX access.log) — filter for unauthenticated POST requests (no valid WordPress nonce or session cookie) targeting /wp-content/plugins/smart-slider-3/ and /wp-admin/admin-ajax.php with HTTP 200 responses between April 7 and the containment timestamp; these represent attacker command-and-control interactions with the backdoor | MySQL binary logs (mysql-bin.*) — reconstruct the exact sequence of INSERT statements used to create rogue administrator accounts and any UPDATE statements modifying wp_options (e.g., cron schedules, siteurl hooks) that the backdoor may have used for secondary persistence; binary logs provide timestamped, row-level evidence that survives file-system cleanup | wp-content/uploads/ directory file listing with inode creation timestamps — the backdoor's unauthenticated remote access capability (CWE-798 hard-coded credential path) likely facilitated web shell upload into the uploads directory; run `find wp-content/uploads/ -name '*.php' -o -name '*.phtml' -o -name '*.php5' | xargs stat` to surface any executable scripts staged for post-exploitation persistence after the initial update-channel compromise

Per-Action IR Details

Step 1: Containment — Immediately disable or take offline any WordPress or Joomla site running Smart Slider 3 Pro v3.5.1.35. Block outbound connections from affected web servers at the firewall/WAF layer. Suspend auto-update functionality for all WordPress/Joomla plugins across your estate until this incident is closed. Sites that auto-updated during April 7 through the vendor patch release should be isolated and treated as compromised.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), NIST CM-3 (Configuration Change Control), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 7.3 (Perform Automated Operating System Patch Management)

Compensating: Run `wp option update auto_update_plugins false` via WP-CLI on each affected host, or add `define('AUTOMATIC_UPDATER_DISABLED', true);` to wp-config.php to halt all auto-updates immediately. At the network layer, use iptables to block egress from the web server IP: `iptables -A OUTPUT -s -j DROP` — restrict to known-good destinations only. For Joomla, disable the Update component under Extensions > Manage and apply the same firewall egress block. If serving behind NGINX, return a 503 maintenance page immediately by toggling the `try_files` block to redirect all traffic to a static maintenance file.

Evidence: Before taking the site offline, capture a full memory image of the web server process (e.g., `gcore $(pgrep php-fpm)`) to preserve any in-memory backdoor payload loaded by v3.5.1.35. Snapshot the current state of `wp-content/plugins/smart-slider-3/`, `wp-content/mu-plugins/`, and `wp-content/uploads/` with `find . -newer /tmp/ref_date -type f -print0 | xargs -0 ls -la` anchored to April 7 00:00 UTC. Capture the WordPress `wp_users` and `wp_usermeta` database tables with `mysqldump --single-transaction wp_users wp_usermeta > users_snapshot.sql` prior to isolation to preserve evidence of any hidden administrator accounts created by the backdoor.

Step 2: Detection — Audit all WordPress and Joomla installations for Smart Slider 3 Pro version 3.5.1.35 using plugin inventory tools or wp-cli ('wp plugin list --all'). Review wp-content/mu-plugins/ for unrecognized files (T1546 persistence via autoload). Search web server access logs for unauthenticated POST requests to plugin directories. Audit WordPress user tables for unknown administrator accounts (T1136.001). Check file system for recently created .php files in plugin, upload, and mu-plugin directories. Review authentication logs for credential use tied to hard-coded or unknown accounts (CWE-798, T1552).

NIST Phase: Detection Analysis

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

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), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 8.2 (Collect Audit Logs), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run `wp plugin list --all --format=csv | grep smart-slider-3` across all managed sites using WP-CLI in batch mode. For mu-plugin backdoor detection, execute `find wp-content/mu-plugins/ -name '*.php' -newer wp-content/plugins/smart-slider-3/ -ls` to isolate files dropped after or alongside the malicious update. Query Apache/NGINX access logs for unauthenticated POST activity to plugin paths with: `grep -E 'POST /wp-content/plugins/smart-slider|POST /wp-admin/admin-ajax' access.log | awk '$9 != 200' | sort | uniq -c | sort -rn`. For hidden admin detection: `wp user list --role=administrator --fields=user_login,user_registered,user_email | sort -k3` — flag any accounts registered on or after April 7. Use YARA with a rule matching obfuscated PHP eval/base64_decode patterns across the plugin directory tree to catch the trojan payload without requiring a SIEM.

Evidence: Collect NGINX or Apache access logs covering April 7 onward, filtering for POST requests to `/wp-content/plugins/smart-slider-3/` and `/wp-admin/admin-ajax.php` with HTTP 200 responses from unauthenticated sources (no valid session cookie). Export the full `wp_users`, `wp_usermeta`, and `wp_options` tables — the backdoor's hidden admin persistence may abuse `wp_options` serialized data (e.g., `siteurl` hooks or cron entries). Capture file hash manifests (`md5sum wp-content/plugins/smart-slider-3/**/*.php > plugin_hashes.txt`) and diff against the vendor's clean v3.5.1.34 file listing to isolate injected files. For Joomla, collect `#__users` and `#__user_groups` table dumps and review `administrator/components/com_smartslider3/` for unexpected files.

Step 3: Eradication — Update Smart Slider 3 Pro to the vendor-released patched version via the vendor's official channel (confirm version number against the Smart Slider HelpScout advisory before applying). Do not use the auto-update mechanism until vendor confirms update channel integrity is restored. Remove all files associated with the malicious build. Delete any administrator accounts not present prior to April 7. Remove unrecognized files from mu-plugins and upload directories. Restore web root from a known-clean backup predating April 7 if forensic review confirms compromise.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SI-3 (Malicious Code Protection), NIST CM-3 (Configuration Change Control), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Download the vendor-patched Smart Slider 3 Pro zip directly from the vendor's HelpScout advisory URL (do not use wp-admin auto-update). Verify the file hash against the vendor-published SHA-256 before extracting. Install manually via `wp plugin install /path/to/smart-slider-clean.zip --force`. For mu-plugin cleanup, run `find wp-content/mu-plugins/ -type f -name '*.php' -newer -exec rm -v {} \;`. To scrub hidden admin accounts: `wp user list --role=administrator | awk 'NR>1{print $1}' | xargs -I{} wp user get {} --field=user_registered` — delete any account with

a registration timestamp on or after 2026-04-07. Scan remaining PHP files with ClamAV: `clamscan -r wp-content/ --include='*.php' -l clam_report.txt`.

Evidence: Before deleting any files, compute and archive SHA-256 hashes of all Smart Slider 3 Pro plugin files (`find wp-content/plugins/smart-slider-3/ -type f -exec sha256sum {} \; > malicious_build_hashes.txt`) for chain-of-custody and potential law enforcement referral. Preserve a full copy of any files dropped in `mu-plugins/` and `uploads/` alongside their inode creation timestamps (`stat `). Archive the MySQL binary log (`mysqlbinlog mysql-bin.* > binlog_archive.sql`) to reconstruct exactly when the backdoor created rogue admin accounts and what privilege escalation queries were executed.

Step 4: Recovery — After applying the clean build, re-audit user accounts, mu-plugins directory, and file system for residual artifacts. Reset all WordPress/Joomla administrator credentials and application secrets (salts, API keys). Re-enable the site only after file integrity verification against a pre-compromise baseline. Monitor web application logs and authentication events for 30 days post-remediation for signs of re-entry via persisted access (T1546, T1098). Confirm WAF rules are updated to detect web shell access patterns.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST IA-5 (Authenticator Management), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AC-2 (Account Management), CIS 5.2 (Use Unique Passwords), CIS 5.3 (Disable Dormant Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Regenerate WordPress salts immediately using `wp config shuffle-salts` to invalidate any active sessions established by the backdoor. Run `wp user update --user_pass=$(openssl rand -base64 24)` to force-rotate all administrator passwords. Verify file integrity post-restoration by running `wp plugin verify-checksums smart-slider-3` and cross-checking against the vendor's published hash manifest. Deploy a Sigma rule targeting POST requests to `wp-content/uploads/*.php` in your NGINX/Apache log pipeline (Sigma rule `web_shell_access.yml` from SigmaHQ community repo covers this pattern). If no SIEM is available, configure a cron job every 15 minutes: `find wp-content/uploads/ -name '*.php' -newer /tmp/recovery_baseline -exec logger -t wp_alert 'Suspicious PHP in uploads: {}' \;`.

Evidence: After restoration, immediately capture a clean file integrity baseline using `md5deep -r wp-content/ > post_recovery_baseline.txt` so any future reinfection produces a diff. Monitor authentication logs (`/var/log/auth.log` on Linux hosts, WordPress `wp_options` `wp_user_roles`, and Joomla `#__session` table) for any login events using accounts that existed during the compromise window — these may indicate persisted credential reuse via T1078. Retain 30 days of post-recovery WAF and web server access logs as evidence of clear passage before declaring full recovery.

Step 5: Post-Incident — This incident exposes a control gap in plugin auto-update trust. Implement a policy requiring integrity verification and staging-environment testing before auto-updates apply to production. Establish a plugin allowlist with version pinning for business-critical WordPress/Joomla plugins. Add plugin directory file integrity monitoring (FIM) to your SIEM detection coverage. Review your software supply chain risk assessment to include third-party plugin update channels as an attack surface.

NIST Phase: Post Incident

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

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST SA-12 (Supply Chain Protection), NIST CM-3 (Configuration Change Control), 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), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Implement version pinning for Smart Slider 3 Pro and all business-critical plugins by adding `define('FS_METHOD', 'direct');` combined with `add_filter('auto_update_plugin', '__return_false');` in wp-config.php and documenting approved version numbers in a plaintext plugin manifest (`plugin_allowlist.txt`) committed to version control. Deploy AIDE (Advanced Intrusion Detection Environment) or `inotifywait` on `wp-content/plugins/` and

`wp-content/mu-plugins/` to alert on any file creation or modification: `inotifywait -m -r -e create,modify wp-content/plugins/ 2>&1 | tee /var/log/fim_plugins.log`. Conduct a lessons-learned session within 5 business days per NIST 800-61r3 §4 and document the Smart Slider update channel compromise as a supply chain threat scenario in your IR tabletop library.

Evidence: Preserve the complete incident timeline — from April 7 auto-update event through full recovery — including all collected log exports, file hash manifests, database snapshots, and deleted account records. This documentation package satisfies NIST IR-6 (Incident Reporting) requirements and provides the evidence baseline for updating your software supply chain risk assessment to classify third-party WordPress/Joomla plugin update channels as an external dependency with elevated trust-chain risk, consistent with NIST SA-12 (Supply Chain Protection).

Detection Guidance

Query web server access logs for POST requests targeting wp-admin/admin-ajax.php or plugin-specific endpoints originating from unexpected IPs during April 7 through the patch release window. Check wp-content/mu-plugins/ for files not present in version control or pre-compromise backups; mu-plugin autoload (T1546) is a confirmed persistence mechanism. Run 'SELECT user_login, user_registered FROM wp_users WHERE user_registered > 2026-04-06' to surface accounts created after the compromise date. Search for PHP files in wp-content/uploads/ (common web shell drop location, T1505.003). Look for base64-encoded strings or eval() patterns in recently modified plugin files. If you have EDR or FIM on the web server, query for file creation events in plugin and mu-plugin directories between April 7 and remediation. Indicators of Compromise (IOCs): As of initial publication, none have been confirmed as publicly available. Monitor Wordfence and vendor advisories for IOC releases as forensic analysis progresses.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://www.bleepingcomputer.com/news/security/smart-slider-updates-hijacked-to-push-malicious-wordpress-joomla-versions/`	BleepingComputer reporting on the Smart Slider 3 Pro supply chain compromise	HIGH
URL	`https://smartslider.helpscoutdocs.com/article/2144-wordpress-security-advisory-smart-slider-3-pro-3-5-1-35-compromise`	Vendor security advisory for the v3.5.1.35 compromise — check for updated file hashes and patched version details	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1078.003** — Local Accounts
- **T1136** — Create Account
- **T1059.004** — Unix Shell
- **T1195.002** — Compromise Software Supply Chain

- **T1505.003** — Web Shell
- **T1136.001** — Local Account
- **T1546** — Event Triggered Execution
- **T1552** — Unsecured Credentials
- **T1059.006** — Python
- **T1556** — Modify Authentication Process
- **T1098** — Account 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-2** — Account Management
- **AC-6** — Least Privilege
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **CM-2** — Baseline Configuration
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **CM-3** — Configuration Change Control
- **SI-10** — Information Input Validation
- **SR-2** — Supply Chain Risk Management Plan

OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures
- **A03:2021** — Injection
- **A07:2021** — Identification and Authentication Failures
- **A04:2021** — Insecure Design

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.3** — Require MFA for Externally-Exposed Applications
- **7.3** — Perform Automated Operating System Patch Management

- **7.4** — Perform Automated Application Patch Management
- **15.1** — Establish and Maintain an Inventory of Service Providers

ISO-27001-2022

- **A.8.28** — Secure coding
- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information
- **A.5.21** — Managing information security in the ICT supply chain

HIPAA-SECURITY

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

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC9.2** — Manages risks associated with vendors and business partners

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1190	Exploit Public-Facing Application	Initial-Access
T1078.003	Local Accounts	Defense-Evasion
T1136	Create Account	Persistence
T1059.004	Unix Shell	Execution
T1195.002	Compromise Software Supply Chain	Initial-Access
T1505.003	Web Shell	Persistence
T1136.001	Local Account	Persistence
T1546	Event Triggered Execution	Privilege-Escalation
T1552	Unsecured Credentials	Credential-Access
T1059.006	Python	Execution
T1556	Modify Authentication Process	Credential-Access
T1098	Account Manipulation	Persistence

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/smart-slider-updates...	T3
Document Structure Overview | PDF | Computing - Scribd	https://www.scribd.com/document/874796808/Filenames-or-Directories-All	T3
Alle Tags: Balou - Clickets	https://www.clickets.de/alltags/balou	T3
WordPress security advisory: Smart Slider 3 Pro 3.5.1.35 compromise	https://smartslider.helpscoutdocs.com/article/2144-wordpress-securi...	T3
Smart Slider updates hijacked to push malicious WordPress, Joomla ...	https://www.bleepingcomputer.com/news/security/smart-slider-updates...	T3

Generated 2026-04-10 06:15 UTC by TJS Security Command Center