INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-15 19:00 UTC

FunnelKit Checkout Plugin Under Active Attack: Card Skimmer Targets 40,000+ WooCommerce Sites

THREAT CAMPAIGN | **CRITICAL** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0320
Type	Threat Campaign
Severity	CRITICAL
CVSS Base Score	7.5
Affected Products	FunnelKit Funnel Builder for WooCommerce Checkout plugin, all versions before 3.15.0.3; WooCommerce sites using this plugin (40,000+ estimated installations)
Published	2026-05-15T15:30:33
Discovery Source	Rss

Executive Summary

An unpatched critical vulnerability in the FunnelKit Funnel Builder WordPress plugin is being actively exploited to inject card-skimming code into WooCommerce checkout pages across an estimated 40,000+ installations. Attackers write malicious JavaScript directly into plugin settings without authentication, capturing payment card numbers, CVVs, and billing data from customers at the moment of purchase. Sites still running versions before 3.15.0.3 are actively exposing customer payment data; sites already compromised may remain infected even after patching unless skimmer code is explicitly removed.

Technical Analysis

Affected product: FunnelKit Funnel Builder for WooCommerce Checkout, all versions before 3.15.0.3. The vulnerability stems from an unprotected checkout endpoint that performs no authentication or capability checks before writing attacker-supplied content to the plugin's External Scripts configuration setting. Relevant weaknesses: CWE-306 (Missing Authentication for Critical Function), CWE-862 (Missing Authorization), CWE-79 (Stored XSS). Exploitation is unauthenticated and remote. Attackers inject JavaScript that harvests payment card numbers, CVVs, and billing fields from checkout page visitors, then exfiltrates data to attacker-controlled infrastructure over WebSocket connections (T1071.001). MITRE coverage: T1190 (Exploit Public-Facing Application), T1059.007 (JavaScript execution), T1565.001 (Stored Data Manipulation), T1027 (Obfuscated Files or Information), T1041 (Exfiltration Over C2 Channel), T1056.003 (Web Portal Capture). A patch was released May 14, 2026 (version 3.15.0.3). CVSS base score: 7.5. A related CVE, CVE-2025-54750,

affecting FunnelKit appears in SentinelOne's vulnerability database; its precise mapping to this unauthenticated endpoint vector is medium-confidence pending authoritative CVE assignment. Active exploitation corroborated by BleepingComputer and Wordfence threat intelligence. CISA KEV listing: not confirmed as of this writing.

Action Checklist

1. Containment, Immediately update FunnelKit Funnel Builder for WooCommerce Checkout to version 3.15.0.3 or later via the WordPress plugin dashboard or wp-cli. If immediate patching is not possible, disable the plugin until the update is applied. Block unauthenticated POST requests to FunnelKit checkout endpoints at the WAF or reverse proxy.
2. Detection, Review the FunnelKit plugin's External Scripts configuration setting in the WordPress admin panel for any JavaScript not added by your team. Scan site files and the WordPress database for injected script tags containing WebSocket connection strings (ws:// or wss://) or Base64-encoded payloads. Query server access logs for unexpected POST requests to FunnelKit checkout endpoints from unfamiliar IPs. Use a WordPress integrity scanner (e.g., Wordfence, WPScan) to detect modified plugin files.
3. Eradication, Apply the official patch (version 3.15.0.3) from the WordPress plugin repository. After patching, manually audit and clear the External Scripts setting, removing any JavaScript not explicitly authorized. Scan the full WordPress database for injected script content. Rotate WordPress admin credentials and WooCommerce API keys as a precaution if compromise is confirmed.
4. Recovery, After cleanup, perform a full checkout transaction test to confirm payment flows function normally and no injected scripts remain active. Monitor outbound network connections from the web server for anomalous WebSocket traffic to external hosts. Re-scan with Wordfence or equivalent within 24 hours of remediation. Verify plugin version is confirmed at 3.15.0.3 or above.
5. Post-Incident, This incident exposes a control gap: unauthenticated write access to script injection surfaces in third-party plugins. Review all other installed WooCommerce and checkout-related plugins for similar endpoint authorization weaknesses. Implement a change management process requiring review of External Scripts and custom code fields on checkout plugins. Evaluate WAF rules for unauthenticated write requests to plugin admin endpoints as a standing control.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to legal, privacy counsel, and PCI DSS QSA if web server access logs confirm unauthenticated POST requests to the FunnelKit endpoint occurred while the plugin was active, as this establishes a confirmed card data exposure window triggering PCI DSS v4.0 Requirement 12.10 incident response obligations and potential breach notification requirements to payment brands, acquiring banks, and affected cardholders.

Recovery Notes	Post-remediation monitoring must focus specifically on the FunnelKit External Scripts database option and outbound WebSocket connections from the web server, as the attacker's injection mechanism writes directly to WordPress database options rather than the filesystem — file integrity monitoring alone will not detect reinfection. Monitor WAF logs for resumed unauthenticated POST attempts to FunnelKit checkout endpoints for a minimum of 30 days, as active exploitation campaigns frequently retry previously successful targets. Confirm with your payment processor that no fraudulent transactions sourced from the compromised checkout page have been flagged, and preserve WooCommerce order records from the exposure window for potential cardholder notification.
Forensic Artifacts	wp_options table row containing the FunnelKit External Scripts option: the injected card-skimmer JavaScript payload is written directly to this database field by unauthenticated attackers — export with 'wp option get funnelkit_external_scripts' and hash before any cleanup. | Web server access logs (nginx: /var/log/nginx/access.log; Apache: /var/log/apache2/access.log) filtered for unauthenticated POST requests to FunnelKit checkout or admin-ajax endpoints — source IPs, timestamps, and request sizes identify the attacker's injection events and establish the breach window. | Outbound network traffic logs or tcpdump PCAPs showing WebSocket upgrade handshakes (HTTP 101 Switching Protocols with 'Upgrade: websocket' header) from the web server to external IPs — these represent the exfiltration channel through which captured card numbers, CVVs, and billing data were transmitted to attacker-controlled infrastructure. | Browser-rendered checkout page source (curl -A 'Mozilla/5.0' https://yoursite.com/checkout > checkout_source_$(date +%F).html) captured before remediation — confirms whether the injected script was being served to customers and provides the exact skimmer code as customers experienced it. | WooCommerce order records (wp_posts and wp_postmeta tables filtered by post_type='shop_order' and post_date within the confirmed injection window) — identifies the population of customer transactions potentially exposed during active skimming, required for PCI DSS breach scope determination and cardholder notification.

Per-Action IR Details

Containment — Immediately update FunnelKit Funnel Builder for WooCommerce Checkout to version 3.15.0.3 or later via the WordPress plugin dashboard or wp-cli. If immediate patching is not possible, disable the plugin until the update is applied. Block unauthenticated POST requests to the FunnelKit checkout endpoint at the WAF or reverse proxy layer.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: prioritize actions that stop ongoing harm before full eradication is possible, particularly when customer PII/PCI data is actively being exfiltrated at point-of-sale.

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SC-7 (Boundary Protection), CIS 7.4 (Perform Automated Application Patch Management), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: If patching is blocked by a change freeze, run: 'wp plugin deactivate woocommerce-funnel-builder --path=/var/www/html' via WP-CLI to disable immediately. At the nginx or Apache layer, add a rule to return 403 on unauthenticated POST requests matching '/wp-admin/admin-ajax.php?action=funnelkit*' or the specific FunnelKit checkout REST endpoint. For Apache: 'RewriteCond %{REQUEST_METHOD} POST' + 'RewriteRule ^/checkout/funnelkit(.*)$ - [F,L]'. Verify the block is active by running 'curl -X POST https://yoursite.com/[endpoint]' and confirming a 403 response.

Evidence: Before patching or disabling the plugin, preserve: (1) a full database dump ('wp db export pre-patch-$(date +%F).sql') to capture the injected JavaScript payload stored in FunnelKit's External Scripts option row in the wp_options table; (2) a filesystem snapshot of wp-content/plugins/woocommerce-funnel-builder/ to preserve any modified plugin PHP files; (3) WAF or reverse proxy access logs showing unauthenticated POST requests to FunnelKit endpoints in the period preceding discovery — these source IPs represent the attacker's injection infrastructure.

Detection — Review the FunnelKit plugin's External Scripts configuration setting in the WordPress admin panel for any JavaScript not added by your team. Scan site files and the WordPress database for injected script tags containing WebSocket connection strings (ws:// or wss://) or Base64-encoded payloads. Query server access logs for unexpected POST requests to FunnelKit checkout endpoints from unfamiliar IPs. Use a WordPress integrity scanner (e.g., Wordfence, WPScan) to detect modified plugin files.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate multiple indicator types (database anomalies, network artifacts, file integrity deltas) to determine scope of card-skimmer injection and whether exfiltration is ongoing.

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-2 (Event Logging), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Run this WP-CLI command to dump the FunnelKit External Scripts option directly: 'wp option get funnelkit_external_scripts --path=/var/www/html' (substitute the actual option key used by the plugin version). To scan the full database for WebSocket strings: 'wp db query "SELECT option_name, option_value FROM wp_options WHERE option_value LIKE \'%ws://%\' OR option_value LIKE \'%wss://%\' OR option_value LIKE \'%atob(%\';" . For Base64 detection across the filesystem: 'grep -rPo "atob\(['\"]([A-Za-z0-9+/=]{40,})['\"]\)" /var/www/html/wp-content/plugins/woocommerce-funnel-builder/'. Analyze web server access logs with: 'grep -E "POST.*/funnelkit|POST.*checkout" /var/log/nginx/access.log | awk \'{print $1}\' | sort | uniq -c | sort -rn' to surface repeated unauthenticated POST source IPs.

Evidence: Capture before any cleanup: (1) the raw value of the FunnelKit External Scripts database option (wp_options row) containing the injected skimmer JavaScript — this is the primary payload artifact; (2) web server access logs (e.g., /var/log/nginx/access.log or /var/log/apache2/access.log) filtered for POST requests to FunnelKit checkout routes in the window prior to detection — look for IPs making repeated unauthenticated POSTs with no session cookie; (3) outbound network connection logs or tcpdump captures showing WebSocket handshakes (Upgrade: websocket headers) from the web server to external IPs, which indicate active exfiltration of captured card data.

Eradication — Apply the official patch (version 3.15.0.3) from the WordPress plugin repository. After patching, manually audit and clear the External Scripts setting, removing any JavaScript not explicitly authorized. Scan the full WordPress database for injected script content. Rotate WordPress admin credentials and WooCommerce API keys as a precaution if compromise is confirmed.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove all components of the threat — the injected skimmer payload in the database, the vulnerable plugin version enabling reinjection, and any credentials that may have been captured or leveraged during the attack window.

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST IA-5 (Authenticator Management), CIS 5.2 (Use Unique Passwords), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Update via WP-CLI: 'wp plugin update woocommerce-funnel-builder --path=/var/www/html' then verify: 'wp plugin get woocommerce-funnel-builder --field=version'. To clear injected scripts from the database without relying on the admin UI: 'wp option update funnelkit_external_scripts \'\'' (replace with the confirmed option key). For a full database scan targeting skimmer patterns: 'wp db search \' Settings > Advanced > REST API. Verify plugin file integrity post-patch using: 'wp plugin verify-checksums woocommerce-funnel-builder'.

Evidence: Before clearing the External Scripts field, export and hash the injected payload for forensic retention: 'wp option get funnelkit_external_scripts > skimmer_payload_$(date +%F).txt && sha256sum skimmer_payload_$(date +%F).txt'. This preserved payload supports threat intelligence sharing (e.g., submission to Wordfence Threat Intelligence or CISA), payment brand notification (Visa/Mastercard require payload evidence for PCI DSS incident reporting), and future YARA rule development targeting this skimmer's WebSocket exfiltration pattern.

Recovery — After cleanup, perform a full checkout transaction test to confirm payment flows function normally and no injected scripts remain active. Monitor outbound network connections from the web server

for anomalous WebSocket traffic to external hosts. Re-scan with Wordfence or equivalent within 24 hours of remediation. Verify plugin version is confirmed at 3.15.0.3 or above.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore systems to verified clean operation, confirm integrity of the checkout payment flow, and maintain heightened monitoring for reinfection attempts targeting the same FunnelKit plugin endpoint.

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 8.2 (Collect Audit Logs)

Compensating: Run a browser-based checkout test using browser developer tools (F12 > Network tab) and filter for WebSocket connections (WS filter) during a test purchase — any ws:// or wss:// connection to a non-payment-processor domain is a reinfection indicator. For server-side outbound monitoring without a SIEM, run: 'ss -tp | grep ESTAB | grep :443' combined with periodic 'tcpdump -i eth0 -w /tmp/outbound_$(date +%F_%H%M).pcap -G 3600 &' capturing one-hour rolling PCAPs for 48 hours post-remediation. Schedule a cron job for nightly database integrity checks: '0 2 * * * wp option get funnelkit_external_scripts --path=/var/www/html >> /var/log/funnelkit_audit.log 2>&1'. Verify plugin version via WP-CLI: 'wp plugin get woocommerce-funnel-builder --fields=name,version,status'.

Evidence: During recovery validation, capture: (1) browser developer tools HAR file from a test checkout transaction confirming no WebSocket connections to external hosts other than legitimate payment processors (Stripe, PayPal, etc.); (2) the output of 'wp option get funnelkit_external_scripts' post-cleanup confirming the field is empty or contains only authorized scripts — timestamp and preserve this as the clean-state baseline; (3) a 24-hour segment of web server access logs post-remediation showing the absence of unauthenticated POST requests to FunnelKit endpoints, establishing a clean traffic baseline for future anomaly detection.

Post-Incident — This incident exposes a control gap: unauthenticated write access to script injection surfaces in third-party plugins. Review all other installed WooCommerce and checkout-related plugins for similar endpoint authorization weaknesses. Implement a change management process requiring review of External Scripts and custom code fields on checkout plugins. Evaluate WAF rules for unauthenticated write requests to plugin admin endpoints as a standing control.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned review focused on the authorization control gap that permitted unauthenticated writes to the FunnelKit script injection surface, and translate findings into durable WAF rules, plugin review procedures, and PCI DSS notification obligations.

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-2 (Flaw Remediation), NIST SI-10 (Information Input Validation), NIST CM-2 (Baseline Configuration), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: Audit all active WooCommerce and checkout-adjacent plugins for unauthenticated REST API or admin-ajax endpoints using WPScan: 'wpscan --url https://yoursite.com --enumerate ap --plugins-detection aggressive' — review any plugin exposing POST endpoints that accept script or code input without authentication. As a standing WAF compensating control in ModSecurity (free, open source), add a rule blocking unauthenticated POSTs to wp-admin/admin-ajax.php where the referrer is absent or the request lacks a valid WordPress nonce pattern. For change management on a budget, create a monthly cron job that exports all FunnelKit and WooCommerce plugin option values containing 'script' to a versioned log file, enabling diff-based change detection: 'wp db query "SELECT option_name, option_value FROM wp_options WHERE option_name LIKE \'%script%\' OR option_name LIKE \'%funnelkit%\'" > /var/log/plugin_options_$(date +%F).log'.

Evidence: For post-incident documentation and regulatory notification: (1) compile the full timeline of unauthenticated POST requests to the FunnelKit endpoint from web server access logs, establishing the earliest possible injection date — this is the breach window start date required for PCI DSS incident reporting to your acquiring bank and payment brands; (2) preserve the extracted skimmer payload with its SHA-256 hash as evidence for submission to Wordfence Threat Intelligence, WPScan vulnerability database, or CISA if the campaign is not yet publicly attributed; (3) document which customer checkout transactions occurred during the confirmed injection window from WooCommerce order logs

(/wp-content/uploads/woocommerce_uploads/ or WooCommerce > Orders filtered by date range) — this defines the population of potentially affected cardholders for breach notification obligations under PCI DSS v4.0 Requirement 12.10.

Detection Guidance

Check the FunnelKit External Scripts configuration field in the WordPress admin for any JavaScript you did not add, particularly any containing WebSocket calls (ws://, wss://), obfuscated strings, Base64 blobs, or references to external domains. In server access logs, filter POST requests to paths containing 'funnel-builder' or FunnelKit-specific checkout endpoints from IPs not associated with your admin team. In WordPress database tables (wp_options or plugin-specific tables), search for stored script content with keywords: WebSocket, eval(, atob(, fromCharCode. Monitor outbound network connections from your web server for WebSocket traffic to unfamiliar external IPs or domains, this is the exfiltration channel. Wordfence's threat intelligence feed (source corroborated) includes signatures for this campaign; ensure Wordfence definitions are current if deployed. No confirmed IOC IPs or domains are available in the source data at this time.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`No confirmed IOC URLs available in source data`	Exfiltration occurs over attacker-controlled WebSocket endpoints; specific domains or IPs have not been confirmed in the available source material	LOW

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1059.007** — JavaScript
- **T1565.001** — Stored Data Manipulation
- **T1071.001** — Web Protocols
- **T1027** — Obfuscated Files or Information
- **T1041** — Exfiltration Over C2 Channel
- **T1056.003** — Web Portal Capture

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

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-7** — Continuous Monitoring
- **IA-2** — Identification and Authentication (Organizational Users)
- **AC-3** — Access Enforcement
- **SI-10** — Information Input Validation
- **IR-5** — Incident Monitoring

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures
- **A01:2021** — Broken Access Control
- **A03:2021** — Injection

CIS-V8

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

ISO-27001-2022

- **A.8.28** — Secure coding
- **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
T1190	Exploit Public-Facing Application	Initial-Access
T1059.007	JavaScript	Execution
T1565.001	Stored Data Manipulation	Impact
T1071.001	Web Protocols	Command-And-Control
T1027	Obfuscated Files or Information	Defense-Evasion
T1041	Exfiltration Over C2 Channel	Exfiltration
T1056.003	Web Portal Capture	Collection

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/funnel-builder-wordp...	T3
FunnelKit – Funnel Builder for WooCommerce Checkout <= 3.15.0.1	https://www.wordfence.com/threat-intel/vulnerabilities/wordpress-pl...	T3
FunnelKit – Funnel Builder for WooCommerce Checkout - WPScan	https://wpscan.com/plugin/funnel-builder/	T3
CVE-2025-54750: FunnelKit Funnel Builder LFI Vulnerability	https://www.sentinelone.com/vulnerability-database/cve-2025-54750/	T3
Multiple Plugins By FunnelKit <= (Various Versions) – Authenticated ...	https://wpscan.com/vulnerability/7b642b6b-58eb-4b17-9d9a-8d58ae74ddcc/	T3

Generated 2026-05-15 19:00 UTC by TJS Security Command Center