INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-07-06 06:34 UTC

Open Redirect Vulnerability in kirilkirkov Ecommerce-CodeIgniter-Bootstrap (CVE-2026-14632)

CVE VULNERABILITY | MEDIUM | CVSS 6.1

SCC Item ID	SCC-CVE-2026-0390
Type	CVE Vulnerability
CVE ID	CVE-2026-14632
Severity	MEDIUM
CVSS Base Score	6.1
EPSS Score	0.0027 (19th percentile)
Affected Products	kirilkirkov Ecommerce-CodeIgniter-Bootstrap up to commit 95dfa8cebbb87ab46ae450643a07241274a74dce
Published	2026-07-04
Discovery Source	Gemini

Executive Summary

An open redirect vulnerability (CVE-2026-14632) has been disclosed in the open-source Ecommerce-CodeIgniter-Bootstrap project maintained by kirilkirkov. The flaw allows an attacker to craft a malicious URL hosted on the legitimate application domain that silently redirects users to attacker-controlled sites, enabling phishing and credential harvesting campaigns. Organizations running this e-commerce application with internet-facing deployments should assess exposure promptly, though CISA has not added this to the Known Exploited Vulnerabilities catalog and severity is rated medium.

Technical Analysis

CVE-2026-14632 is an open redirect vulnerability (CWE-601) in kirilkirkov's Ecommerce-CodeIgniter-Bootstrap affecting the `href` argument within `application/core/MY_Controller.php`. All versions up to and including commit 95dfa8cebbb87ab46ae450643a07241274a74dce are affected. The vulnerability allows a remote, unauthenticated attacker to manipulate the redirect destination parameter, causing the application to forward users to arbitrary external URLs without validation. This enables abuse of the trusted application domain as a redirect proxy, commonly used as a phishing lure delivery mechanism (MITRE ATT&CK T1566, Phishing) or to socially engineer users into executing malicious content (T1204, User Execution). CVSS base score is 6.1 (medium). EPSS score is 0.00273 (19th percentile), indicating low current exploitation probability. A public

exploit is reported to exist per source data. This CVE is not listed on the CISA KEV catalog. NVD and VulDB listings are cited as primary references. No vendor CVSS vector or official patch identifier was available in the source data.

Action Checklist

1. Step 1: Containment. Identify any internet-facing instances of kirilkirkov Ecommerce-CodeIgniter-Bootstrap at or before commit 95dfa8cebbb87ab46ae450643a07241274a74dce. If a WAF is in place, create a rule to block requests where the `href` redirect parameter contains an external domain not on an approved allowlist, or alert on such requests for manual review.
2. Step 2: Detection. Review web server and application logs for requests to endpoints handled by `application/core/MY_Controller.php` where the `href` parameter value contains an external URL (i.e., begins with `http://` or `https://` pointing to a domain outside your own). Flag any such requests for user notification review. No specific event IDs are available for this application; log query must be written against your web server access logs (Apache/Nginx) or application-level logging if enabled. No IOCs (IPs, domains, hashes) were present in the source data.
3. Step 3: Eradication. Apply input validation and allowlist-based redirect controls within `application/core/MY_Controller.php` to restrict the `href` argument to internal paths or an explicit set of approved domains. No official vendor patch or tagged release was identified in the source data; remediation requires direct code review and fix against the affected commit. Check the kirilkirkov Ecommerce-CodeIgniter-Bootstrap GitHub repository for any commits after 95dfa8cebbb87ab46ae450643a07241274a74dce that address input validation on the `href` parameter; if a fix commit exists, apply it or cherry-pick the relevant changes. Reference the project repository at the identified commit for diff comparison.
4. Step 4: Recovery. After applying the code fix, validate that redirect functionality only resolves to approved internal paths. Test with a crafted external URL as the `href` value to confirm the fix blocks the redirect. Monitor application logs for continued open redirect attempts for at least 30 days post-remediation. Confirm no user accounts show signs of credential compromise resulting from prior phishing redirection.
5. Step 5: Post-Incident. Review whether open redirect validation (NIST AC-4, Information Flow Enforcement; OWASP redirect and forward validation guidance) is part of your secure development lifecycle and pre-deployment code review checklist. Assess whether CIS Safeguard 7.1 (Establish and Maintain a Vulnerability Management Process) is being applied to open-source dependencies and community projects in your application stack, not only commercial software.

IR / Forensic Enrichment

Triage Priority	STANDARD
Escalation Criteria	Escalate to urgent and initiate breach notification assessment if log review confirms that users were successfully redirected to external phishing domains via CVE-2026-14632, resulting in potential credential harvesting — particularly if the application processes payment card data (PCI-DSS) or personal information subject to state breach notification laws.

Recovery Notes	After applying the allowlist-based redirect fix to `application/core/MY_Controller.php`, validate with active testing using crafted external `href` values before returning the application to full production traffic. Monitor Apache/Nginx access logs for `href=https?://` patterns targeting external domains for a minimum of 30 days post-remediation, as threat actors who previously distributed phishing URLs leveraging this domain may continue sending victims to the now-patched endpoint. If any user accounts are identified as having followed attacker-crafted redirect URLs during the exploitation window, treat those accounts as potentially compromised and enforce password resets prior to recovery sign-off.
Forensic Artifacts	Apache/Nginx web server access logs (`/var/log/apache2/access.log` or `/var/log/nginx/access.log`): primary artifact class — contains raw `href` parameter values submitted to `MY_Controller.php` redirect endpoints, source IPs of users who followed attacker-crafted URLs, and timestamps correlating to the phishing campaign window | CodeIgniter application logs (`application/logs/log-YYYY-MM-DD.php`): if `$config['log_threshold']` > 0, may contain controller-level routing records capturing the `href` argument value as passed into the open redirect code path in `MY_Controller.php` | Application database `ci_sessions` table: contains active and recent session records with source IP and user agent for authenticated users — cross-reference session creation timestamps and IPs against the open redirect abuse window to identify accounts that may have been phished | Web server referrer log fields: HTTP `Referer` headers in access logs may reveal attacker-controlled phishing pages that sent users back to the legitimate application domain after credential harvesting, completing the redirect loop and providing attacker infrastructure indicators | Git repository commit history diff for `application/core/MY_Controller.php` at commit 95dfa8cebbb87ab46ae450643a07241274a74dce: documents the exact vulnerable redirect logic for evidence preservation and confirms the code path exploited, supporting root cause documentation in the post-incident report

Per-Action IR Details

Step 1: Containment — Identify any internet-facing instances of kirilkirkov Ecommerce-CodeIgniter-Bootstrap at or before commit 95dfa8cebbb87ab46ae450643a07241274a74dce. If a WAF is in place, create a rule to block or alert on requests where the `href` redirect parameter contains an external domain not on an approved allowlist.

NIST Phase: Containment

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

Controls: NIST AC-4 (Information Flow Enforcement), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: Without a WAF, use Nginx `map` or Apache `mod_rewrite` to inspect the `href` query parameter and return 403 if the value matches `^https?://` followed by any domain outside your own. Example Nginx snippet: `if ($arg_href ~* "^https?://(?!yourdomain\.com)") { return 403; }`. Run `grep -E 'href=https?://[^/]' /var/log/nginx/access.log` to surface historical exploitation attempts immediately.

Evidence: Before deploying any WAF rule or server-side block that modifies live traffic handling, capture current web server access logs (Apache: `/var/log/apache2/access.log`; Nginx: `/var/log/nginx/access.log`) including any in-memory log buffers not yet flushed to disk. Record active TCP connections via `netstat -ano` or `ss -tnp` to identify any persistent attacker sessions riding on currently open HTTP connections to the application. These are volatile and lost on service restart or rule-triggered connection drops.

Step 2: Detection — Review web server and application logs for requests to endpoints handled by `application/core/MY_Controller.php` where the `href` parameter value contains an external URL (i.e., begins with `http://` or `https://` pointing to a domain outside your own). Flag any such requests for user notification review. No specific event IDs are available for this application; log query must be written against your web

server access logs (Apache/Nginx) or application-level logging if enabled. No IOCs (IPs, domains, hashes) were present in the source data.

NIST Phase: Detection Analysis

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

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Run the following against Apache/Nginx access logs to surface all open redirect abuse attempts against this application: `grep -E 'href=https?://[^%]' /var/log/nginx/access.log | awk '{print $1, $7, $11}' | sort | uniq -c | sort -rn`. For URL-encoded variants, also run: `grep -E 'href=%68%74%74%70' /var/log/nginx/access.log`. Extend lookback to the full retention window — phishing campaigns leveraging this flaw may predate your awareness of CVE-2026-14632 by weeks.

Evidence: No volatile host state is altered by this detection step. However, prior to any log rotation event or log management action, preserve a forensic copy of all web server access logs covering the period up to and including commit 95dfa8cebbb87ab46ae450643a07241274a74dce deployment. Also collect CodeIgniter application-level logs from `application/logs/` (if `$config['log_threshold']` is set above 0) which may capture `href` parameter values passed into `MY_Controller.php` routing logic. These logs are the primary artifact class for this vulnerability — there are no host-level process or memory artifacts from an open redirect exploit.

Step 3: Eradication — Apply input validation and allowlist-based redirect controls within `application/core/MY_Controller.php` to restrict the `href` argument to internal paths or an explicit set of approved domains. No official vendor patch or tagged release was identified in the source data; remediation requires direct code review and fix against the affected commit. Reference the project repository at the identified commit for diff comparison.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Without a formal code review pipeline, perform a targeted diff: `git diff 95dfa8cebbb87ab46ae450643a07241274a74dce -- application/core/MY_Controller.php` to identify all redirect-handling code paths. Implement the allowlist fix manually: in `MY_Controller.php`, add a PHP validation block that checks the `href` value against `parse_url()` and compares the `host` component to an explicit array of approved domains before executing any redirect. After applying, use `curl -v 'https://yourapp.com/redirect?href=https://evil.com'` to confirm the fix returns a non-redirect response (400/403) rather than a 301/302 to the external domain.

Evidence: Before modifying `application/core/MY_Controller.php` or restarting the application server to load the fix, capture: (1) a read-only copy of the current `MY_Controller.php` file with hash (`sha256sum application/core/MY_Controller.php`) for chain-of-custody comparison post-remediation; (2) current web server access logs to preserve the pre-fix exploitation baseline; (3) a snapshot of any active PHP-FPM or application server process list (`ps aux | grep php`) to document runtime state before the service reload that will load the patched controller. No RAM capture is required for this vulnerability class — open redirect exploits leave no persistent memory artifacts on the server.

Step 4: Recovery — After applying the code fix, validate that redirect functionality only resolves to approved internal paths. Test with a crafted external URL as the `href` value to confirm the fix blocks the redirect. Monitor application logs for continued open redirect attempts for at least 30 days post-remediation. Confirm no user accounts show signs of credential compromise resulting from prior phishing redirection.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AC-2 (Account Management), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 5.3 (Disable Dormant Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Use `curl -I 'https://yourapp.com/redirect?href=https://attacker.com'` post-fix to confirm HTTP 400/403 is returned rather than 301/302. For account compromise review without SIEM, extract user login records from the CodeIgniter session table (typically `ci_sessions` in the application database) and cross-reference login timestamps and source IPs against the window of identified open redirect abuse in the access logs. Flag any accounts whose last login IP differs from historical baseline and force password reset for those accounts via the application's admin panel.

Evidence: No additional volatile evidence capture is required at this phase, as the eradication step preserved pre-fix artifacts. However, before forcing any account password resets (which terminates active sessions), export the current `ci_sessions` database table: `mysqldump -u [user] -p [dbname] ci_sessions > ci_sessions_prerecovery.sql`. Active session records may contain source IPs and user agent strings that corroborate which user accounts were exposed to attacker-crafted redirect URLs during the exploitation window.

Step 5: Post-Incident — Review whether open redirect validation (NIST AC-4 — Information Flow Enforcement; OWASP redirect and forward validation guidance) is part of your secure development lifecycle and pre-deployment code review checklist. Assess whether CIS Safeguard 7.1 (Establish and Maintain a Vulnerability Management Process) is being applied to open-source dependencies and community projects in your application stack, not only commercial software.

NIST Phase: Post Incident

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

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 2.2 (Ensure Authorized Software is Currently Supported), NIST AU-11 (Audit Record Retention)

Compensating: Add a YARA-style pattern or `grep` scan to your CI/CD pre-commit hook or manual deployment checklist targeting CodeIgniter controller files: `grep -rn 'redirect.*\$.*href\|header.*Location.*\$' application/controllers/ application/core/` to flag any future unvalidated redirect constructs before they reach production. Document kirilkirkov Ecommerce-CodeIgniter-Bootstrap in your software inventory (CIS 2.1) with its GitHub commit hash as the version identifier, since no tagged release versioning exists, enabling future CVE tracking against commit-level exposure.

Evidence: No volatile evidence capture is applicable at this phase. Retain all web server access logs, the pre-fix `MY_Controller.php` hash record, the `ci_sessions` database export, and the WAF/server rule change records for a minimum period consistent with your data retention policy (NIST AU-11) — at least 90 days recommended — to support any downstream regulatory inquiry if user credential compromise from phishing activity is later confirmed.

Detection Guidance

Query web server access logs (Apache, Nginx, or equivalent) for GET or POST requests to routes handled by `application/core/MY_Controller.php` where the `href` parameter (URL-decoded) contains a value beginning with `http://` or `https://` pointing to a domain other than your own application domain. Search for `href=http` or `href=https` in decoded logs, then filter results to exclude your own domain. A more precise regex (after URL decoding) would be `href=https?://(?!yourdomain\.com)[^\s&"]+`, but log format and encoding vary - manual review of candidates is recommended. No specific IOC hashes, IPs, or domains were present in the source data, so detection relies on behavioral pattern matching rather than indicator matching. If your application does not log query parameters, enable parameter-level logging before conducting this review. No SIEM rule signatures specific to this CVE were available in the source data. Map detections to MITRE ATT&CK T1566 (Phishing) as a downstream abuse pattern.

Framework Mappings

MITRE-ATTACK

- **T1566** — Phishing

- **T1204** — User Execution

NIST-800-53R5

- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(5)(i)** — Security Awareness and Training

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1566	Phishing	Initial-Access
T1204	User Execution	Execution

Sources

Source	URL	Tier
gemini	https://www.tenable.com/cve/2026-14632	T1
CVE-2026-14632 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-14632	T1
CVE-2026-14632 in Ecommerce-CodeIgniter-Bootstrap	https://vuldb.com/cve/CVE-2026-14632	T3

Source	URL	Tier
CVE-2026-14632 - Exploits & Severity - Feedly	https://feedly.com/cve/CVE-2026-14632	T3
Known Exploited Vulnerabilities Catalog | CISA	https://www.cisa.gov/known-exploited-vulnerabilities-catalog	T1

Generated 2026-07-06 06:34 UTC by TJS Security Command Center