INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-22 06:47 UTC

CVE-2026-6568: A vulnerability was determined in kodcloud KodExplorer up to 4.52. This affects the function share.c...

CVE VULNERABILITY | HIGH | CVSS 7.3

SCC Item ID	SCC-CVE-2026-0067
Type	CVE Vulnerability
CVE ID	CVE-2026-6568
Severity	HIGH
CVSS Base Score	7.3
EPSS Score	0.0009 (25th percentile)
Affected Products	KodCloud KodExplorer up to version 4.52
Published	2026-04-19T10:16:09.203
Discovery Source	Nvd

Executive Summary

CVE-2026-6568 is a path traversal vulnerability in KodCloud KodExplorer (versions up to 4.52), a self-hosted file management platform. An unauthenticated remote attacker can manipulate a file-sharing parameter to read arbitrary files outside the intended directory, including configuration files, credentials, and other sensitive data stored on the host. No patch is confirmed available, and a public exploit has been disclosed, raising the risk for any organization running this product.

Technical Analysis

CVE-2026-6568 (CWE-22: Path Traversal) affects the share.class.php::initShareOld function within the Public Share Handler component of KodCloud KodExplorer versions up to and including 4.52. The vulnerable file is located at /app/controller/share.class.php. An attacker supplies a crafted 'path' parameter to the share handler, traversing outside the webroot to read arbitrary host filesystem files. The attack requires no authentication and is remotely exploitable over the network. CVSS base score: 7.3 (High). EPSS score: 0.0009 (25th percentile). MITRE ATT&CK mappings: T1190 (Exploit Public-Facing Application), T1083 (File and Directory Discovery). The vendor did not respond to pre-disclosure notification. No patch is currently confirmed. A public exploit has been disclosed.

Action Checklist

1. Containment: Restrict external access to any KodExplorer instance (versions up to 4.52) by blocking public internet exposure at the firewall or WAF. If the service cannot be taken offline, restrict access to trusted IP ranges only.
2. Detection: Review web server access logs for requests to /app/controller/share.class.php containing path traversal sequences (e.g., '../', '%2e%2e%2f', '%252e%252e'). Query SIEM for HTTP requests targeting the share endpoint with anomalous 'path' parameter values. Correlate with file access logs for reads of sensitive files (e.g., /etc/passwd, configuration files) from the web server process.
3. Eradication: No vendor-confirmed patch is available as of this advisory. If upgrade or patch becomes available via the KodCloud project (https://kodcloud.com), apply immediately. Until a patch exists, disable or remove the public share functionality if not operationally required, or take the service offline.
4. Recovery: After applying any available patch or configuration mitigation, verify the 'path' parameter is properly sanitized by testing with traversal sequences in a non-production environment. Monitor web server and file access logs for continued anomalous path requests. Audit files accessible by the web server process and confirm no sensitive data was exposed.
5. Post-Incident: Review whether KodExplorer instances had access to sensitive data or credentials beyond operational necessity. Assess whether least-privilege controls on the web server process would have limited file read exposure. Add KodExplorer to your patch monitoring process for vendor updates.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to legal, privacy, and executive stakeholders immediately if web server access logs confirm successful reads of /etc/passwd, KodExplorer credential stores, or any files containing PII/PHI — triggering breach notification obligations — or if the team lacks the capability to determine whether exploitation occurred within the exposure window.
Recovery Notes	After applying the path-sanitization mitigation or vendor patch, run targeted curl-based validation against the share.class.php endpoint using encoded traversal sequences before returning KodExplorer to production. Monitor web server access logs daily for 30 days for renewed traversal attempts against the share endpoint, particularly from source IPs identified in the initial detection sweep. If confirmed exploitation occurred during the unpatched window, treat all credentials stored in or accessible by the KodExplorer data directory as compromised and rotate them before resuming service.

Forensic Artifacts	Web server access logs (nginx: /var/log/nginx/access.log; Apache: /var/log/apache2/access.log; IIS: %SystemDrive%\inetpub\logs\LogFiles\W3SVC*) — contain the raw HTTP requests to /app/controller/share.class.php with 'path' parameter values showing traversal sequences, source IPs, response codes, and response sizes indicating whether sensitive file contents were returned (large 200 responses to traversal requests are high-confidence indicators of successful exfiltration). | KodExplorer application logs under /data/logs/ — may record share link generation events and file access operations triggered by the vulnerable share controller, providing application-layer correlation to supplement web server logs. | OS-level file read audit events from Linux auditd for reads of /etc/passwd, /etc/shadow, and KodExplorer config files (/config/) by the web server process user — confirm whether traversal payloads successfully reached and read sensitive files beyond the webroot. | KodExplorer configuration files at /config/ and user data at /data/User/ — determine the scope of credentials and user session tokens that were readable via path traversal, required for blast radius assessment and breach notification decisions. | Filesystem modification timestamps on all PHP files under — detect any web shells or backdoors written to the server if the vulnerability allowed write access in addition to read, or if a chained exploit was used during the exposure window.

Per-Action IR Details

Containment — Immediately restrict external access to any KodExplorer instance (versions up to 4.52) by blocking public internet exposure at the firewall or WAF. If the service cannot be taken offline, restrict access to trusted IP ranges only.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy (CSF RS.MA-01: Execute IR plan, contain and mitigate)

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: On Linux hosts: run `iptables -I INPUT -p tcp --dport 80 -j DROP && iptables -I INPUT -p tcp --dport 443 -j DROP` to immediately block inbound web traffic, then add allowlist rules for trusted source IPs via `iptables -I INPUT -s -p tcp --dport 80 -j ACCEPT`. On Windows IIS hosts: use Windows Firewall (`netsh advfirewall firewall add rule name='Block KodExplorer Public' protocol=TCP dir=in localport=80,443 action=block`). If a WAF is unavailable, use nginx `deny all; allow ;` directives in the KodExplorer location block. Document the block timestamp and approved IP allowlist before implementing.

Evidence: Before blocking, capture a snapshot of active connections to the KodExplorer web port: run `ss -tnp sport = :80 or sport = :443` (Linux) or `netstat -anob | findstr :80` (Windows) and save output with timestamp. Preserve the current web server access log (`/var/log/nginx/access.log`, `/var/log/apache2/access.log`, or IIS `%SystemDrive%\inetpub\logs\LogFiles\W3SVC*`) as a read-only forensic copy before any log rotation occurs. Record the KodExplorer process owner and open file handles via `lsof -p ` to establish pre-containment file access baseline.

Detection — Review web server access logs for requests to /app/controller/share.class.php containing path traversal sequences (e.g., '../', '%2e%2e%2f', '%252e%252e'). Query SIEM for HTTP requests targeting the share endpoint with anomalous 'path' parameter values. Correlate with file access logs for reads of sensitive files (e.g., /etc/passwd, configuration files) from the web server process.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis (CSF DE.AE-02: Analyze adverse events; DE.AE-03: Correlate information from multiple sources)

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), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without a SIEM, run this grep against the web server access log to find traversal attempts targeting the share endpoint: `grep -iE '/app/controller/share\.class\.php.*(\.\./|%2e%2e%2f|%252e%252e|%2e%2e/)' /var/log/nginx/access.log | tee /tmp/kod_traversal_hits.txt`. For double-encoded variants, also run: `grep -iE 'share.*path=.*(%25|%2[ef]|%5c)' /var/log/nginx/access.log`. On Linux, correlate with OS-level file read audit events using auditd: `ausearch -k file_read -f /etc/passwd` or enable with `auditctl -w /etc/passwd -p r -k file_read`. Deploy the Sigma rule community detection for generic path traversal (sigma/rules/web/web_path_traversal_exploitation.yml) adapted with the KodExplorer-specific URI pattern. Use `awk '{print $1}' /tmp/kod_traversal_hits.txt | sort | uniq -c | sort -rn` to rank source IPs by request volume.

Evidence: Preserve raw web server access logs with original timestamps and permissions intact before analysis — copy to `/tmp/forensic_kod_$(date +%Y%m%d%H%M%S)/` and hash with `sha256sum`. Extract and preserve any KodExplorer application-level logs stored under the KodExplorer data directory (default: `/data/logs/`). Capture OS audit logs for file reads by the web server process user (www-data, apache, nginx) against `/etc/passwd`, `/etc/shadow`, KodExplorer config files at `/config/`, and `/data/User/`. On Linux, run `find -name '*.php' -newer /var/log/nginx/access.log -ls` to identify any web shells potentially dropped post-traversal.

Eradication — No vendor-confirmed patch is available as of this advisory. If upgrade or patch becomes available via the KodCloud project (https://kodcloud.com), apply immediately. Until a patch exists, disable or remove the public share functionality if not operationally required, or take the service offline.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication (CSF RS.MA-01: Remove threat from environment, verify eradication)

Controls: NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality), NIST SI-5 (Security Alerts, Advisories, and Directives), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Without a patch, disable the vulnerable share controller by making it non-executable: `chmod 000 /app/controller/share.class.php` and verify with `ls -la`. Alternatively, add a blocking location rule in nginx: `location ~* /app/controller/share\.class\.php { return 403; }` and reload nginx (`nginx -s reload`). For Apache, add to `.htaccess` or VirtualHost: ` Require all denied `. Monitor the KodCloud GitHub repository (https://github.com/kalcaddle/KodExplorer) and the NVD entry for CVE-2026-6568 for patch availability, and assign a team member to check weekly. Document this as a temporary mitigation in your risk register with a review date.

Evidence: Before disabling the share controller, preserve a forensic copy of `share.class.php` with hash: `cp /app/controller/share.class.php /tmp/forensic_kod_$(date +%Y%m%d)/share.class.php && sha256sum /tmp/forensic_kod_*/share.class.php`. Capture the full KodExplorer directory tree with timestamps: `find -printf '%T+ %m %u %g %p\n' | sort > /tmp/kod_filetree_pre_eradication.txt`. If exploitation is confirmed, run `find /tmp /var/tmp /data -name '*.php' -o -name '*.phtml' | xargs md5sum` to baseline all PHP files and detect potential web shells placed via traversal-enabled file write (confirm whether the vulnerability is read-only or enables write, and scope evidence accordingly).

Recovery — After applying any available patch or configuration mitigation, verify the 'path' parameter is properly sanitized by testing with traversal sequences in a non-production environment. Monitor web server and file access logs for continued anomalous path requests. Audit files accessible by the web server process and confirm no sensitive data was exposed.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery (CSF RC: Execute recovery plan, restore systems, verify integrity, communicate)

Controls: NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-2 (Flaw Remediation), NIST CA-7 (Continuous Monitoring), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Validate path sanitization in a non-production KodExplorer instance using curl: `curl -v 'http:///app/controller/share.class.php?path=../../etc/passwd'` and variants `%2e%2e%2f`, `%252e%252e%2f`, and `....//` — a patched version should return 403/400 or a permission error, not file contents. Use a YARA rule to scan the KodExplorer data directory for credential patterns: `yara -r /data/` where the rule matches strings like `root:x:0:0`

(passwd content) or KodExplorer config keys that may have been exfiltrated and cached. Set up a cron job to run `diff ` daily for 30 days post-recovery to detect any late-stage persistence dropped during the exploitation window.

Evidence: Before returning to production, generate a new integrity baseline of the KodExplorer installation: `find -type f -exec sha256sum {} \; > /tmp/kod_post_recovery_baseline.txt` and compare against the pre-incident baseline if available. Confirm the web server process user (e.g., `www-data`) cannot read files outside the KodExplorer webroot: `sudo -u www-data cat /etc/passwd` should fail. Preserve post-patch web server access logs as a clean baseline for 90 days, tagged with the recovery timestamp, to support retrospective analysis if delayed indicators of compromise emerge.

Post-Incident — Review whether KodExplorer instances had access to sensitive data or credentials beyond operational necessity. Assess whether least-privilege controls on the web server process would have limited file read exposure. Add KodExplorer to your patch monitoring process for vendor updates.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity (CSF GV/ID: Lessons learned, update policies, improve detection, share intelligence)

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-2 (Flaw Remediation), NIST AC-6 (Least Privilege), NIST RA-3 (Risk Assessment), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 3.2 (Establish and Maintain a Data Inventory)

Compensating: Run `find / -user www-data -readable -not -path '/*' 2>/dev/null` to enumerate files readable by the web server process outside the application directory — document findings to scope the blast radius. Review `/etc/passwd` and `/etc/group` to confirm the web server process user has no shell (`/usr/sbin/nologin`) and no supplementary group memberships granting database or config file access. Add a cron-driven check using `rpm -q kodexplorer` or file version inspection against the KodCloud changelog to alert on version changes. Subscribe to the KodCloud GitHub releases page (https://github.com/kalcaddle/KodExplorer/releases) via RSS for patch notifications without requiring a SIEM.

Evidence: Compile a lessons-learned artifact package including: (1) the preserved web server access logs showing traversal attempts with source IPs and timestamps, (2) the file tree diff showing any unexpected modifications during the exposure window, (3) the list of files readable by the web server process user, and (4) the timeline from public CVE-2026-6568 disclosure to containment. This package supports breach notification assessment if regulated data (PII, PHI, credentials) was confirmed readable via the traversal path. Archive all forensic hashes and command outputs with analyst attribution per NIST AU-10 (Non-Repudiation) requirements.

Detection Guidance

Search web server access logs (Apache, nginx, or IIS) for requests to /app/controller/share.class.php. Flag any 'path' parameter values containing: '../', '..%2f', '%2e%2e', '%252e%252e', or absolute path prefixes (e.g., '/etc/', '/var/', 'C:\Windows'). In a SIEM, create a rule for HTTP 200 responses to the share endpoint combined with traversal-pattern parameters; a 200 response with traversal sequences indicates successful exploitation. Also monitor OS-level file access logs (auditd on Linux) for reads of sensitive files (e.g., /etc/passwd, /etc/shadow, application config files) by the web server process (www-data, apache, nginx). No confirmed IOC hashes or IPs are available from current sources.

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1083** — File and Directory Discovery

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-3** — Access Enforcement
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

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

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **16.12** — Implement Code-Level Security Checks
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain
- **A.5.23** — Information security for use of cloud services

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1190	Exploit Public-Facing Application	Initial-Access
T1083	File and Directory Discovery	Discovery

Sources

Source	URL	Tier
nvd	https://nvd.nist.gov/vuln/detail/CVE-2026-6568	T1
(consolidated)	https://nvd.nist.gov/vuln/detail/CVE-2026-6569	T1

Source	URL	Tier
CVE-2026-6568: A vulnerability was determined Directory trav	https://www.sherlockforensics.com/blog/2026-04-20-cve-2026-6568.html	T3
A vulnerability was determined in kodcloud KodExplorer up ...	https://github.com/advisories/GHSA-rgfh-mp7v-25f9	T3
CVE-2026-6568 - Exploits & Severity	https://feedly.com/cve/CVE-2026-6568	T3

Generated 2026-04-22 06:47 UTC by TJS Security Command Center