INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-20 06:45 UTC

CVE-2026-8734: A vulnerability was determined in Oinone Pamirs up to 7.2.0. Affected by this issue is the function ...

CVE VULNERABILITY | HIGH | CVSS 7.3

SCC Item ID	SCC-CVE-2026-0197
Type	CVE Vulnerability
CVE ID	CVE-2026-8734
Severity	HIGH
CVSS Base Score	7.3
EPSS Score	0.0003 (8th percentile)
Affected Products	Oinone Pamirs up to version 7.2.0
Published	2026-05-17T06:16:19.490
Discovery Source	Nvd

Executive Summary

A SQL injection vulnerability in Oinone Pamirs (versions up to and including 7.2.0) allows remote attackers to execute arbitrary database queries through a publicly exposed interface. The exploit has been publicly disclosed, and the vendor did not respond to prior notification, meaning no official patch is currently available. Organizations running this platform should treat it as unpatched and act on mitigation controls immediately.

Technical Analysis

CVE-2026-8734 is a SQL injection vulnerability (CWE-89, CWE-74) in the RSQLToSQLNodeConnector.makeVariable function, exposed through the queryListByWrapper interface in Oinone Pamirs up to and including version 7.2.0. A remote, unauthenticated or low-privilege attacker can manipulate input to this interface to inject arbitrary SQL statements against the underlying database. The attack vector maps to MITRE ATT&CK T1190 (Exploit Public-Facing Application). CVSS base score is 7.3 (High). EPSS score is 0.028% (0.08th percentile), indicating minimal current exploitation activity in the wild, though the exploit is publicly disclosed. No vendor patch exists as of this report. No CISA KEV listing as of this report.

Action Checklist

1. Prevention & Isolation: Identify all instances of Oinone Pamirs version 7.2.0 or earlier in your environment. If the queryListByWrapper interface is internet-facing, immediately restrict access via WAF rule or firewall ACL blocking external access to that endpoint until remediation is complete.

2. Detection & Monitoring: Review web application and database logs for anomalous SQL syntax in query parameters directed at the queryListByWrapper endpoint, including patterns such as single quotes, UNION SELECT, OR 1=1, stacked queries, and comment sequences (-- or /**/). Query your SIEM for T1190-mapped events against this application host.

3. Mitigation & Compensating Controls: No official vendor patch is available. Apply input validation and parameterized query controls at the application layer if you have source access. If not, enforce WAF rules blocking SQL metacharacters on affected endpoints. Consider taking the affected interface offline if business impact permits.

4. Validation: After applying WAF or network controls, validate that the queryListByWrapper endpoint returns expected errors or is blocked for injected input. Run a baseline database integrity check to confirm no unauthorized data access or modification occurred during the exposure window.

5. Post-Mitigation Review: Document the gap: no vendor patch was available at time of exposure. Evaluate your vendor risk management process for products where vendors are unresponsive to security notifications. Add Oinone Pamirs to your continuous monitoring inventory and reassess if a patch is released.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO, legal counsel, and data protection officer immediately if database forensic review reveals SELECT queries against PII, PHI, or credential tables during the exposure window, or if the blast radius extends to more than one internet-facing Oinone Pamirs instance — active public disclosure of CVE-2026-8734 with no vendor patch elevates breach notification risk under GDPR, HIPAA, and state privacy statutes.
Recovery Notes	Do not restore external access to the queryListByWrapper endpoint until WAF block validation testing against SQLi payloads is documented and passed. Monitor database query logs and web server access logs continuously for at least 14 days post-containment, as attackers who successfully enumerated data during the exposure window may return via alternative vectors or with session artifacts. Reassess compensating control adequacy every 30 days until an official Oinone Pamirs patch is released and deployed, given the unpatched-by-vendor status of CVE-2026-8734.

Forensic Artifacts	Web server access logs (nginx/access.log or Apache access_log) containing raw URL-encoded requests to the queryListByWrapper endpoint — preserve with original encoding intact, as percent-encoded SQLi payloads (%27, %20UNION%20SELECT%20) are direct exploitation indicators specific to this HTTP-exposed interface. | MySQL/MariaDB general query log or binary log (binlog) covering the exposure window — SQL injection against Oinone Pamirs' queryListByWrapper would manifest as structurally anomalous queries originating from the Pamirs application DB user, including UNION-based column enumeration, tautology patterns (OR 1=1), or stacked semicolon-delimited statements absent from normal ORM output. | Application-layer error logs from the Oinone Pamirs Java runtime (typically under /opt/pamirs/logs/ or equivalent) — successful or near-successful SQL injection attempts in Java-based ORM frameworks frequently generate SQLException stack traces that reveal which queries were executed, including attacker-injected fragments. | WAF or reverse proxy decision logs (if ModSecurity or equivalent is present) — these capture the full request body and URI at the point of interception and provide timestamped evidence of attack attempts against the queryListByWrapper endpoint both before and after rule deployment. | Database information_schema snapshot — a point-in-time export of 'SELECT table_name, table_rows, update_time FROM information_schema.tables' for the Pamirs database captures whether attacker-driven SELECT, INSERT, UPDATE, or DELETE operations altered row counts or modification timestamps on sensitive tables during the CVE-2026-8734 exposure window.

Per-Action IR Details

Containment — Identify all instances of Oinone Pamirs version 7.2.0 or earlier in your environment. If the queryListByWrapper interface is internet-facing, immediately restrict access via WAF rule or firewall ACL blocking external access to that endpoint until remediation is complete.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolate affected systems to prevent further exploitation while preserving evidence and maintaining business continuity where possible.

Controls: NIST IR-4 (Incident Handling), NIST SI-10 (Information Input Validation), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Run 'grep -r "pamirs" /etc/hosts /opt /var/www 2>/dev/null' and check package manifests or Docker image labels to enumerate Oinone Pamirs deployments. Use iptables to block external access: 'iptables -I INPUT -p tcp --dport -s 0.0.0.0/0 ! -s -j DROP'. For WAF-less environments, use Nginx 'deny all;' in the location block for /queryListByWrapper and reload the config. Document each identified instance with IP, hostname, and version in a running incident log.

Evidence: Before restricting access, capture a full snapshot of active network connections to the Oinone Pamirs application port using 'ss -tnp' or 'netstat -tnp' and preserve the output timestamped. Collect current web server access logs (e.g., /var/log/nginx/access.log or Apache's access_log) covering the full exposure window to preserve pre-containment traffic. Record the application version from deployment manifests or the Pamirs admin console before any changes are made.

Detection — Review web application and database logs for anomalous SQL syntax in query parameters directed at the queryListByWrapper endpoint, including patterns such as single quotes, UNION SELECT, OR 1=1, stacked queries, and comment sequences (-- or //). Query your SIEM for T1190-mapped events against this application host.**

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate log sources to determine scope, confirm exploitation, and characterize attacker activity against the specific vulnerable endpoint.

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-3 (Content of Audit Records), NIST AU-12 (Audit Record Generation), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, run this grep against web server access logs to surface SQL injection attempts against the specific endpoint: 'grep -i "queryListByWrapper" /var/log/nginx/access.log | grep -iE "(union|select|insert|update|delete|drop|--|/*|or\s+1=1|'+)"'. For database-layer visibility on MySQL/MariaDB, enable the general query log temporarily: 'SET GLOBAL general_log = ON; SET GLOBAL general_log_file="/tmp/mysql_general.log";' — then grep for queries sourced from the Pamirs application user containing UNION, stacked semicolons, or ORDER BY position-based injection patterns. Use the free Sigma rule for T1190 (Exploit Public-Facing Application) converted to grep/awk for flat log analysis.

Evidence: Collect the full web server access log for the queryListByWrapper URI path, preserving raw URL-encoded query strings — do not normalize or decode before preservation, as encoding variations (e.g., %27 for single quote, %20UNION%20) are themselves forensic indicators. Capture the database slow query log and general query log covering the exposure window; SQL injection probes often generate anomalous query shapes (e.g., tautologies, UNION-based column enumeration) that differ structurally from legitimate Pamirs ORM-generated queries. Preserve any application-layer error logs that may contain stack traces revealing successful or partially successful injection attempts.

Eradication — No official vendor patch is available. Apply input validation and parameterized query controls at the application layer if you have source access. If not, enforce WAF rules blocking SQL metacharacters on affected endpoints. Consider taking the affected interface offline if business impact permits.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the vulnerability or reduce exploitability to an acceptable level; in the absence of a vendor patch, compensating controls at the WAF and network layer constitute the eradication action.

Controls: NIST SI-2 (Flaw Remediation), NIST SI-10 (Information Input Validation), NIST SI-3 (Malicious Code Protection), NIST CM-6 (Configuration Settings), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: For teams with ModSecurity (free, open source WAF), enable the OWASP CRS ruleset and specifically activate rules 942100–942999 (SQL Injection Attack) scoped to the queryListByWrapper endpoint. Deploy a custom rule blocking known SQLi metacharacters: 'SecRule ARGS "@detectSQLi" "id:9001,phase:2,deny,status:403,msg:\'SQLi attempt on queryListByWrapper\',chain" SecRule REQUEST_URI "@contains queryListByWrapper"'. If source code is accessible, replace any string-concatenated query construction in the Pamirs queryListByWrapper handler with PreparedStatement (Java) or parameterized query equivalents in the framework's ORM layer. Document the compensating control formally per NIST SI-2 guidance as an accepted risk with a defined review date.

Evidence: Before deploying WAF rules or code changes, capture the exact HTTP request structure (headers, method, body, query parameters) of confirmed or suspected malicious requests to queryListByWrapper from access logs — this preserves the attack signature for future detection tuning. Take a read-only snapshot or export of the affected database tables accessible via this endpoint to establish a pre-eradication baseline for later integrity comparison. If the application is containerized, export the running container image ('docker commit') before any changes to preserve the exploitable state as forensic evidence.

Recovery — After applying WAF or network controls, validate that the queryListByWrapper endpoint returns expected errors or is blocked for injected input. Run a baseline database integrity check to confirm no unauthorized data access or modification occurred during the exposure window. Restore normal operations only after validation.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: verify that compensating controls are effective, confirm environment integrity, and restore only after evidence-based validation — not time-based assumption.

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST CP-10 (System Recovery and Reconstitution), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Validate WAF efficacy by replaying known-safe SQLi test payloads (from SQLMap's tamper library or manual crafting) against the queryListByWrapper endpoint from an authorized test client and confirm HTTP 403 or connection reset responses — log results with timestamps. For database integrity, run 'SELECT table_name, table_rows, update_time FROM information_schema.tables WHERE table_schema="" and compare row counts and last-modified timestamps against a pre-incident baseline if available; flag any tables modified during the exposure window. Use 'pt-table-checksum' (free Percona tool) for deeper row-level integrity verification on MySQL/MariaDB without requiring downtime.

Evidence: Before restoring full external access, preserve post-control web server access logs to confirm no SQLi pattern traffic is passing through. Export database audit logs (MySQL binary log / general log) covering the period from initial exposure to containment, as these may be required for breach notification analysis — specifically look for SELECT queries against user, credential, or PII-bearing tables that originated from the Pamirs application database user during anomalous hours. Document WAF rule test results (blocked vs. allowed payloads) as evidence that compensating controls are functioning before sign-off.

Post-Incident — Document the gap: no vendor patch was available at time of exposure. Evaluate your vendor risk management process for products where vendors are unresponsive to security notifications. Add Oinone Pamirs to your continuous monitoring inventory and reassess if a patch is released.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned, update policies and monitoring based on incident findings, and feed intelligence back into the preparation phase to reduce recurrence risk.

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST IR-6 (Incident Reporting), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST RA-3 (Risk Assessment), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Create a tracked exception entry in your vulnerability register for CVE-2026-8734 noting: unpatched status, compensating controls applied, review date (recommend 30 days), and vendor responsiveness rating. Set a cron-based or RSS-feed alert on the NVD entry for CVE-2026-8734 to catch any future patch publication: 'curl -s "https://services.nvd.nist.gov/rest/json/cves/2.0?cveId=CVE-2026-8734" | jq ".vulnerabilities[0].cve.metrics"' run weekly. Flag Oinone Pamirs in your software inventory (CIS 2.1) with a 'vendor-unresponsive' tag to trigger elevated scrutiny in future vendor risk reviews.

Evidence: Compile and retain the complete incident record including: timeline from first exposure to containment, all log extracts collected during investigation, WAF rule deployment records, database integrity check results, and the absence-of-patch documentation (screenshot or export of vendor advisory page or disclosure timeline). This package constitutes the evidentiary record required if the exposure window involved regulated data (PII, PHI, financial records) and breach notification obligations must be assessed. Retain per your organization's documented retention schedule per NIST AU-11 (Audit Record Retention).

Detection Guidance

Monitor web server and application logs for requests to the queryListByWrapper endpoint containing SQL injection patterns: single quotes ('), double dashes (--), UNION SELECT, OR conditions (e.g., OR 1=1), and URL-encoded equivalents (%27, %20OR%20). At the database layer, enable query logging and alert on unexpected SELECT, INSERT, UPDATE, or DROP statements originating from the application service account outside normal business logic. In your SIEM, correlate application-layer anomalies with unusual database read volume or access to tables outside normal application scope. No public IOCs (IPs, domains, hashes) have been confirmed for active exploitation campaigns at this time.

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application

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-10** — Information Input Validation

OWASP-TOP10-2021

- **A03:2021** — Injection

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures

ISO-27001-2022

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

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

Sources

Source	URL	Tier
nvd	https://nvd.nist.gov/vuln/detail/CVE-2026-8734	T1
CVE-2026-8734 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-8734	T3
CVE-2026-8734: Oinone Pamirs up to SQL injection - Sherlock	https://www.sherlockforensics.com/blog/2026-05-17-cve-2026-8734.html	T3

Source	URL	Tier
CVE-2026-24660 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-24660	**T3**
CVE-2026-8734 | INCIBE-CERT	https://www.incibe.es/en/incibe-cert/early-warning/vulnerabilities/...	**T3**

Generated 2026-05-20 06:45 UTC by TJS Security Command Center