INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-27 06:39 UTC

CVE-2026-39830: golang.org/x/crypto/ssh Client-Induced Server Deadlock in Microsoft Azure Linux cert-manager

CVE VULNERABILITY | **CRITICAL** | CVSS 9.1

SCC Item ID	SCC-CVE-2026-0225
Type	CVE Vulnerability
CVE ID	CVE-2026-39830
Severity	CRITICAL
CVSS Base Score	9.1
EPSS Score	0.0004 (13th percentile)
Affected Products	Microsoft azl3 cert-manager 1.12.15-6 on Azure Linux 3.0 (golang.org/x/crypto/ssh)
Published	2026-05-27T01:40:27
Discovery Source	Msrc Patch Tuesday

Executive Summary

A critical vulnerability in the golang.org/x/crypto/ssh package allows a malicious SSH client to force a server-side deadlock, causing denial of service. Microsoft's cert-manager package (version 1.12.15-6) on Azure Linux 3.0 is confirmed affected, with a CVSS score of 9.1. Organizations running Kubernetes workloads on Azure Linux 3.0 that depend on cert-manager for TLS certificate lifecycle management face potential disruption to certificate issuance and renewal, which can cascade into service outages.

Technical Analysis

CVE-2026-39830 is a deadlock vulnerability (CWE-833: Deadlock, CWE-667: Improper Locking) in golang.org/x/crypto/ssh. An SSH client can send unexpected responses during an established session, triggering a server-side deadlock in the SSH handler goroutine. The affected package is golang.org/x/crypto/ssh as bundled in Microsoft's azl3 cert-manager 1.12.15-6 on Azure Linux 3.0. CVSS base score is 9.1 (Critical); EPSS is 0.042% (score: 0.00042, 12.75th percentile), indicating low observed exploitation activity at time of disclosure. MITRE techniques T1499 (Endpoint Denial of Service) and T1499.004 (Application or System Exploitation) apply. No CISA KEV listing as of disclosure. The vulnerability is not exploitation-for-code-execution; the impact is availability loss through process deadlock. Vendor advisory: MSRC May 2026 Patch Tuesday (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-39830). NVD entry: https://nvd.nist.gov/vuln/detail/CVE-2026-39830. Before beginning remediation, verify patched package

availability from the MSRC advisory. If a patch is not yet available, focus on containment pending vendor release.

Action Checklist

1. Step 1: Containment, Identify all Azure Linux 3.0 nodes running cert-manager 1.12.15-6 using 'rpm -q cert-manager' or equivalent package query. Implement network segmentation to allow SSH connections to cert-manager pods only from designated internal management hosts. Block all external SSH access at the network perimeter (CIS 4.4, 4.5). If no patch is available and exposure is broad, consider disabling SSH-dependent cert-manager workflows and routing certificate requests through an alternate method.
2. Step 2: Detection, Query package inventory for azl3 cert-manager 1.12.15-6 across all Azure Linux 3.0 nodes. Review Kubernetes pod logs for cert-manager for deadlock indicators: look for hung SSH handler processes, unresponsive certificate signing requests, or cert-manager controller logs showing stalled reconciliation loops. Run 'kubectl logs -c cert-manager' and search for goroutine dumps or 'blocked on' patterns. Check system-level audit logs (NIST AU-2, AU-6) for anomalous SSH connection attempts against cert-manager endpoints. There are no public IOCs associated with active exploitation of this CVE at time of disclosure.
3. Step 3: Eradication, Verify the patched cert-manager package and version are available from the MSRC May 2026 Patch Tuesday advisory (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-39830) before proceeding. Once available, update via the Azure Linux package manager ('tdnf update cert-manager' on Azure Linux 3.0). Confirm the updated golang.org/x/crypto/ssh dependency version is included in the patched build (CIS 7.3, 7.4).
4. Step 4: Recovery, After patching, restart cert-manager pods and validate certificate issuance and renewal workflows end-to-end. Confirm TLS certificates for critical services have not expired during any disruption window. Monitor cert-manager controller logs for resumed reconciliation and absence of deadlock indicators (NIST SI-4, AU-6). Verify Kubernetes cluster health metrics return to baseline.
5. Step 5: Post-Incident, Review SSH exposure surface for all Kubernetes workload components, not only cert-manager (CIS 7.1). Assess whether cert-manager SSH handling was a known risk in your asset inventory (CIS 1.1, 2.1). Implement automated patch compliance monitoring to detect future instances of unpatched critical packages on Azure Linux nodes (CIS 7.3, 7.4). Document this event per NIST IR-5 incident tracking requirements.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to senior IR leadership and notify cloud infrastructure ownership immediately if cert-manager deadlock is confirmed active (stalled CertificateRequests observed), if any TLS certificates for PCI-DSS, HIPAA, or SOC 2 scoped services have expired or are within 24 hours of expiry due to the disruption, or if SSH connections to cert-manager endpoints from untrusted external sources are identified in audit logs indicating active exploitation rather than unintentional triggering.

Recovery Notes	After patching and pod restart, monitor cert-manager controller logs continuously for a minimum of 72 hours for recurrence of goroutine deadlock patterns, as intermittent malformed SSH client connections could re-trigger the condition if network access controls were not fully enforced during the containment phase. Prioritize immediate manual renewal of any TLS certificates that expired during the disruption window by running 'kubectl annotate certificate -n cert-manager.io/issue-temporary-certificate=true' as a bridge measure. Validate that all dependent services relying on cert-manager-issued certificates have successfully re-established valid TLS sessions before closing the incident, paying particular attention to ingress controllers and internal service mesh mTLS configurations that may have silently failed over to unencrypted or self-signed fallback behavior.
Forensic Artifacts	cert-manager controller pod logs (/var/log/containers/cert-manager-*.log on the node, or via 'kubectl logs') containing goroutine stack traces, SSH handler hang messages, or reconciliation loop stall timestamps specific to the deadlock triggered by CVE-2026-39830 | Azure Linux 3.0 auditd log (/var/log/audit/audit.log) CRYPTO_SESSION and USER_AUTH event records timestamped against cert-manager process PID activity, identifying the source IP and timing of SSH connections that induced the deadlock condition | Kubernetes CertificateRequest and CertificateSigningRequest object history ('kubectl get csr -A -o yaml') showing Pending requests that stalled during the deadlock window, establishing the service impact timeline and identifying which workloads lost certificate lifecycle management | Pre- and post-patch RPM package metadata ('rpm -qi cert-manager' and 'rpm -q --requires cert-manager') documenting the vulnerable golang.org/x/crypto/ssh dependency version present in cert-manager 1.12.15-6 and confirming its replacement in the patched build | tdnf transaction history ('tdnf history' or '/var/log/tdnf.log') recording the exact timestamp, package version, and node identity for the cert-manager patch application, serving as the chain-of-custody record for eradication verification

Per-Action IR Details

Step 1: Containment — Identify all Azure Linux 3.0 nodes running cert-manager 1.12.15-6 using 'rpm -q cert-manager' or equivalent package query. Restrict SSH access to cert-manager pods to trusted internal management hosts only; block external SSH reach to cert-manager services at the network perimeter (CIS 4.4, CIS 4.5). If exposure is broad and a patch is not yet applied, consider temporarily disabling SSH-dependent cert-manager workflows and routing certificate requests through an alternate method.

NIST Phase: Containment

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

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: Run 'kubectl get pods -A -o wide | grep cert-manager' across all clusters to enumerate affected pods and their node assignments, then cross-reference with 'rpm -q cert-manager' via SSH or Azure Run Command on each node. Use 'iptables -I INPUT -p tcp --dport 22 -s -j ACCEPT && iptables -I INPUT -p tcp --dport 22 -j DROP' on affected nodes to restrict SSH to management hosts only. Document every node FQDN and pod name before applying network controls — this is your containment scope baseline.

Evidence: Before restricting SSH access, capture the current list of active SSH connections to cert-manager nodes using 'ss -tnp | grep :22' and 'who -a' on each node. Collect 'kubectl logs -n cert-manager --previous' to preserve any pre-containment deadlock indicators. Snapshot current cert-manager CertificateRequest and CertificateSigningRequest object states via 'kubectl get csr -A -o yaml > csr_snapshot_$(date +%Y%m%d%H%M%S).yaml' to establish a pre-containment baseline of certificate pipeline health.

Step 2: Detection — Query package inventory for azl3 cert-manager 1.12.15-6 across all Azure Linux 3.0 nodes. Review Kubernetes pod logs for cert-manager for goroutine deadlock indicators: look for hung SSH

handler processes, unresponsive certificate signing requests, or cert-manager controller logs showing stalled reconciliation loops. Check system-level audit logs (NIST AU-2, AU-6) for anomalous SSH connection attempts against cert-manager endpoints. There are no public IOCs associated with active exploitation of this CVE at time of disclosure.

NIST Phase: Detection Analysis

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

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Query all Azure Linux 3.0 nodes for the vulnerable package: 'for node in $(kubectl get nodes -o jsonpath="{.items[*].metadata.name}"); do echo $node; kubectl debug node/$node -it --image=mcr.microsoft.com/cbl-mariner/base/core:2.0 -- rpm -q cert-manager; done'. For deadlock detection without a SIEM, run 'kubectl logs -n cert-manager -l app=cert-manager --since=24h | grep -iE "deadlock|goroutine|timeout|stuck|stall|ssh"' and pipe through 'tee cert_manager_deadlock_$(date +%Y%m%d).log'. For SSH anomaly detection on the node, review '/var/log/audit/audit.log' filtering on 'type=USER_AUTH' and 'type=NET_SESSION' events correlated with cert-manager process PIDs.

Evidence: Collect cert-manager controller pod logs in full via 'kubectl logs -n cert-manager deployment/cert-manager --since=72h > cert_manager_controller_72h.log' before log rotation discards goroutine stack traces. On each Azure Linux 3.0 node, preserve '/var/log/audit/audit.log' entries showing SSH connection attempts (auditd event type CRYPTO_SESSION and USER_AUTH) timestamped against cert-manager process activity. Run 'kubectl get events -n cert-manager --sort-by=.lastTimestamp > cert_manager_events.log' to capture Kubernetes-level evidence of CertificateRequest failures or controller reconciliation timeouts caused by the deadlock condition.

Step 3: Eradication — Apply the updated cert-manager package published by Microsoft for Azure Linux 3.0 as documented in the MSRC May 2026 Patch Tuesday advisory (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-39830). Verify the patched package version against the MSRC advisory before deployment. Update via the Azure Linux package manager ('tdnf update cert-manager' on Azure Linux 3.0). Confirm the updated golang.org/x/crypto/ssh dependency version is included in the patched build (CIS 7.3, CIS 7.4).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-3 (Configuration Change Control), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Before patching, verify the vulnerable dependency is present: 'rpm -q --requires cert-manager | grep golang-x-crypto' or inspect the binary directly with 'strings /usr/bin/cert-manager | grep x/crypto'. After running 'tdnf update cert-manager' on each Azure Linux 3.0 node, confirm eradication with 'rpm -q --changelog cert-manager | head -30' to verify the CVE-2026-39830 fix is listed, and 'rpm -qi cert-manager' to record the exact post-patch version for your change record. If automated patching via tdnf is not available, download the patched RPM from the Azure Linux package repository and install with 'rpm -Uvh cert-manager-.azl3.rpm' after verifying its SHA-256 checksum against the MSRC advisory.

Evidence: Before applying the patch, capture 'rpm -qi cert-manager' and 'rpm -q --requires cert-manager' output to document the pre-patch state of the vulnerable package and its golang.org/x/crypto/ssh dependency version. Preserve a copy of the cert-manager binary ('cp /usr/bin/cert-manager /forensics/cert-manager.pre-patch.$(date +%Y%m%d)') for potential future forensic analysis. After patching, re-run the same queries and record the post-patch versions as evidence of successful remediation, storing both outputs together in your incident record per NIST IR-5 (Incident Monitoring) documentation requirements.

Step 4: Recovery — After patching, restart cert-manager pods and validate certificate issuance and renewal workflows end-to-end. Confirm TLS certificates for critical services have not expired during any disruption window. Monitor cert-manager controller logs for resumed reconciliation and absence of deadlock indicators (NIST SI-4, AU-6). Verify Kubernetes cluster health metrics return to baseline.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST CP-10 (System Recovery and Reconstitution), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Restart cert-manager with 'kubectl rollout restart deployment/cert-manager -n cert-manager' and watch rollout status via 'kubectl rollout status deployment/cert-manager -n cert-manager'. Validate certificate pipeline recovery by checking all CertificateRequest objects clear from Pending: 'kubectl get certificaterequests -A | grep -v Approved | grep -v Denied'. Identify any TLS certificates that expired during the disruption window using 'kubectl get certificates -A -o jsonpath="{range .items[*]}{.metadata.name}{\t}{.status.notAfter}{\n}{end}" | awk -v now="$(date -u +%Y-%m-%dT%H:%M:%SZ)" "$2 < now"'. Set a 72-hour log watch with 'kubectl logs -n cert-manager -l app=cert-manager -f | grep -iE "error|deadlock|ssh|timeout"' to confirm sustained recovery.

Evidence: At recovery start, document the exact timestamp of pod restart and the pre-restart state of all CertificateRequest objects. After cert-manager resumes, collect 'kubectl get certificates -A -o yaml > certificates_post_recovery_$(date +%Y%m%d%H%M%S).yaml' to establish a post-recovery certificate health baseline and identify any services that experienced TLS disruption during the deadlock window — this record supports both incident documentation and downstream service impact assessment.

Step 5: Post-Incident — Review SSH exposure surface for all Kubernetes workload components, not only cert-manager (CIS 7.1). Assess whether cert-manager SSH handling was a known risk in your asset inventory (CIS 1.1, CIS 2.1). Implement automated patch compliance monitoring to detect future instances of unpatched critical packages on Azure Linux nodes (CIS 7.3, CIS 7.4). Document this event per NIST IR-5 incident tracking requirements.

NIST Phase: Post Incident

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

Controls: NIST IR-5 (Incident Monitoring), NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Enumerate all Kubernetes workload components that expose SSH services using 'kubectl get pods -A -o json | jq ".items[] | select(.spec.containers[].ports[]?.containerPort == 22) | {namespace: .metadata.namespace, pod: .metadata.name}"'. For ongoing patch compliance monitoring on Azure Linux 3.0 without enterprise tooling, create a cron job that runs 'tdnf check-update --security 2>&1 | tee /var/log/tdnf_security_check_$(date +%Y%m%d).log' nightly and emails output to the security team. To detect future instances of the golang.org/x/crypto/ssh vulnerability class across your Go-based workloads, deploy an osquery scheduled query: 'SELECT name, version FROM rpm_packages WHERE name LIKE "%golang-x-crypto%" OR name LIKE "%cert-manager%"' and baseline against known-good versions.

Evidence: Compile the full incident timeline from collected artifacts: pre-patch 'rpm -qi' outputs, cert-manager log extracts showing deadlock indicators, containment timestamps from network change records, patch application records from 'tdnf history', and post-recovery certificate health snapshots. This package constitutes the incident record required by NIST IR-5 (Incident Monitoring) and should be retained per NIST AU-11 (Audit Record Retention) policy; it also feeds the lessons-learned review to determine whether cert-manager's SSH dependency was tracked in your software inventory under CIS 2.1 (Establish and Maintain a Software Inventory).

Detection Guidance

No active exploitation IOCs are publicly associated with CVE-2026-39830 at time of disclosure (EPSS 0.042%, not on CISA KEV). Detection focus is on identifying vulnerable instances and signs of deadlock-induced denial of service. Check for: (1) cert-manager 1.12.15-6 present on Azure Linux 3.0 nodes via package query ('rpm -qa | grep cert-manager'); (2) cert-manager pod logs showing stalled or non-completing certificate signing requests;

(3) cert-manager pod logs showing goroutine stack traces with blocked SSH handler threads (indicative of deadlock). To inspect: run 'kubectl logs -c cert-manager' and search for goroutine dumps or 'blocked on' patterns; (4) Kubernetes events showing cert-manager controller restarts or unresponsive states; (5) failed TLS certificate renewals surfacing as downstream service errors. Audit log coverage per NIST AU-2 and AU-12 should include cert-manager pod stdout/stderr and Kubernetes audit logs for controller activity. Per NIST AU-6, review these logs at defined intervals for anomaly patterns consistent with T1499.004 (application denial of service via resource exhaustion or deadlock). Until vendor-specific detection rules are published, rely on infrastructure-level monitoring: alert on pod restarts for cert-manager, certificate issuance delays exceeding SLA thresholds, and SSH connection attempts to cert-manager endpoints from unexpected sources.

Framework Mappings

MITRE-ATTACK

- **T1499** — Endpoint Denial of Service
- **T1499.004** — Application or System Exploitation

NIST-800-53R5

- **SC-5** — Denial-of-Service Protection
- **SC-13** — Cryptographic Protection

CIS-V8

- **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.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1499	Endpoint Denial of Service	Impact
T1499.004	Application or System Exploitation	Impact

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-39830	T1
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-May	T1

Source	URL	Tier
CVE-2026-39830 - CVE Details, Severity, and Analysis | Strobes VI	https://strobes.co/vi/cve/CVE-2026-39830/	T3
Linux Distros Unpatched Vulnerability : CVE-2026-39830 | Tenable®	https://www.tenable.com/plugins/nessus/316570	T3
CVE-2026-33430: Briefcase Privilege Escalation Vulnerability	https://www.sentinelone.com/vulnerability-database/cve-2026-33430/	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-39830	T1

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-05-27 06:39 UTC by TJS Security Command Center