INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-03-29 18:39 UTC

Cisco IOS XR CLI Privilege Escalation and Command Injection Vulnerabilities Allow Root-Level Code Execution

SECURITY ANALYSIS | **HIGH** | CVSS 7.8

SCC Item ID	SCC-CVE-2026-0008
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.8
Affected Products	Cisco IOS XR Software (multiple versions; see Cisco Security Advisories cisco-sa-iosxr-priv-esc-GFQjxvOF and cisco-sa-xr-cmdinj-vsKGherc for affected release lists)
Published	11 hours ago
Discovery Source	Rss

Executive Summary

Cisco disclosed two high-severity vulnerabilities in IOS XR Software affecting enterprise and service provider routers. Both flaws require authenticated local access but allow an attacker to escalate privileges to root on the underlying operating system, granting complete device control. Organizations running IOS XR in core routing infrastructure should treat patching as a priority, as successful exploitation could cause sustained network disruption or enable persistent footholds in high-availability environments.

Technical Analysis

Two distinct vulnerability classes affect Cisco IOS XR Software. First, a CLI privilege escalation flaw (cisco-sa-iosxr-priv-esc-GFQjxvOF, CWE-269) allows an authenticated local attacker with restricted CLI access to escalate to root on the underlying OS. Second, a command injection vulnerability (cisco-sa-xr-cmdinj-vsKGherc, CWE-78) allows an authenticated local attacker to inject and execute arbitrary OS-level commands as root. Both carry a CVSS base score of 7.8. Attack vector is local; authentication is required. MITRE ATT&CK relevance: T1078 (Valid Accounts), T1059 (Command and Scripting Interpreter), T1068 (Exploitation for Privilege Escalation). CVE identifiers were not confirmed in source data, consult the Cisco Security Advisory pages directly for confirmed CVE IDs and the full list of affected IOS XR releases. Cisco has released patches; no fully mitigating workarounds are confirmed. No active exploitation or threat actor attribution has been publicly established. Note: CVE ID fields and EPSS data were absent from source input and are not estimated here.

Action Checklist

1. Step 1, Patch: Review Cisco Security Advisories cisco-sa-iosxr-priv-esc-GFQjxvOF and cisco-sa-xr-cmdinj-vsKGherc to identify affected IOS XR releases; apply Cisco-provided patches on all affected devices following your change management process.
2. Step 2, Inventory: Identify all IOS XR devices in your environment (enterprise routers, service provider edge, core routing infrastructure) and cross-reference running software versions against the affected release lists in both advisories.
3. Step 3, Access Review: Audit which accounts hold local CLI access to IOS XR devices; restrict access to the minimum required set and enforce least-privilege role assignments pending patch deployment.
4. Step 4, Detection: Review authentication and command execution logs on IOS XR devices for unexpected privilege changes, anomalous local logins, or unusual OS-level command execution patterns (see detection guidance).
5. Step 5, Communication: Notify network operations, change management, and relevant stakeholders of the patching timeline; escalate to leadership if patching critical routing infrastructure requires a maintenance window that affects availability SLAs.
6. Step 6, Long-Term: Review local access provisioning controls for network devices; evaluate whether privileged access management (PAM) tooling or jump-host enforcement reduces exposure surface for similar local-access vulnerabilities in future.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	If any affected IOS XR device is in a critical routing path (carries production traffic to multiple customer sites or core network segments) and patching requires more than a 4-hour maintenance window or cannot be tested in a staging environment first, escalate to VP Network Ops and Chief Information Security Officer with risk assessment and business continuity impact analysis.
Recovery Notes	After patching is complete and devices are confirmed stable (verify 'show version' reflects new build, confirm 'show processes' and interface status are nominal, test routing protocol adjacency recovery), run a post-patch health check: execute baseline commands ('show bgp summary', 'show interface', 'show memory'), compare against pre-patch logs to identify any degradation. Document the final patched state in the CMDB. Schedule a post-incident review with network and security teams to capture lessons learned (e.g., patch deployment process improvements, monitoring enhancements).
Forensic Artifacts	/var/log/messages (syslog: command execution, authentication events, system state changes) | /var/log/secure (auth log: login attempts, privilege escalation attempts, account modifications) | /var/log/audit/audit.log (auditd: system calls, setuid/setgid execution, file access if enabled) | IOS XR 'show users' output (active CLI sessions, source IP, privilege level) | IOS XR 'show running-config' exports (account provisioning, AAA configuration, role assignments, baseline for detecting unauthorized modifications)

Per-Action IR Details

Step 1, Patch: Review Cisco Security Advisories cisco-sa-iosxr-priv-esc-GFQjxvOF and cisco-sa-xr-cmdinj-vsKGherc to identify affected IOS XR releases; apply Cisco-provided patches on all affected devices following your change management process.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (Preparation phase: tools, techniques, and processes)

Controls: NIST 800-53 SI-2 (Flaw Remediation), CIS 3.10 (Address Unauthorized Software)

Compensating: If change management is manual, document the current IOS XR version on each device using 'show version' CLI output, save to a timestamped file (e.g., ios-xr-baseline-2026-03-04.txt), verify Cisco advisory applicability by matching release string, then schedule patches during maintenance windows with documented approval. Use free Cisco Software Checker (web-based) to validate version eligibility before patching.

Evidence: Capture pre-patch CLI output: 'show version', 'show running-config', 'show processes', 'show users' (to document active sessions). Export to plain text. Preserve /var/log/messages and /var/log/secure on IOS XR (Linux-based XR) to establish baseline for post-patch comparison. Document patch application timestamps in change ticket.

Step 2, Inventory: Identify all IOS XR devices in your environment (enterprise routers, service provider edge, core routing infrastructure) and cross-reference running software versions against the affected release lists in both advisories.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (asset inventory and management)

Controls: NIST 800-53 CM-8 (Information System Component Inventory), CIS 1.1 (Inventory and Control of Enterprise Assets)

Compensating: If CMDB is unavailable, use a spreadsheet: query each router via SSH with 'show version | include Version' and 'show inventory' (hardware models), pipe output to a CSV (hostname, model, IOS XR version, site). Cross-reference manually against the two Cisco advisories' affected release tables. For air-gapped or hard-to-reach devices, document serial numbers and last-known software version; escalate those separately. Use free tools: Nmap (port 22/830 for NETCONF) to discover potential IOS XR devices, then validate with CLI.

Evidence: Export pre-inventory baseline: SNMP walks (sysDescr OID .1.3.6.1.2.1.1.1.0), 'show version' output from each device saved to timestamped log file. Preserve network topology diagrams showing routing dependencies. Document which devices are in active traffic paths (for containment planning).

Step 3, Access Review: Audit which accounts hold local CLI access to IOS XR devices; restrict access to the minimum required set and enforce least-privilege role assignments pending patch deployment.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (access control provisioning); NIST 800-53 AC-2 (Account Management)

Controls: NIST 800-53 AC-2 (Account Management), NIST 800-53 AC-6 (Least Privilege), CIS 5.2 (Use Unique Passwords)

Compensating: If no centralized AAA system exists, manually audit each IOS XR device: 'show aaa users' (active sessions) and 'show running-config | include username' (provisioned accounts). Export to a text file; identify accounts that should not exist (test accounts, departed staff). Revoke unnecessary accounts using 'no username '. For local accounts, enforce role-based access via 'usergroup' commands (e.g., 'usergroup netops' with limited tasks). Document the least-privilege baseline per role. Use SSH key-based auth if available; disable password auth where possible.

Evidence: Capture AAA configuration before and after: 'show running-config | section aaa', 'show aaa users' (active sessions), 'show running-config | section username', 'show privilege' (privilege levels). Export to timestamped config backup. Document who has access and why (correlate to RACI matrix). Preserve any TACACS+/RADIUS logs if available.

Step 4, Detection: Review authentication and command execution logs on IOS XR devices for unexpected privilege changes, anomalous local logins, or unusual OS-level command execution patterns (see detection guidance).

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 (Detection and Analysis); NIST 800-53 AU-12 (Audit Generation)

Controls: NIST 800-53 AU-2 (Audit Events), NIST 800-53 SI-4 (Information System Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: If no SIEM is available, manually collect and parse logs: SSH to each IOS XR device, extract /var/log/messages (syslog), /var/log/secure (auth), /var/log/audit/audit.log (auditd records if present). Use 'grep' to search for privilege escalation keywords: 'sudo', 'su -', 'setuid', 'execve', 'CAP_SYS', and any failed/successful logins outside business hours. Pipe results to a timestamped log file for each device. Cross-correlate timestamps with change tickets. Use free log aggregation: Splunk free tier or Graylog community edition if possible.

Evidence: Preserve in chain-of-custody format: /var/log/messages (CLI commands, syslog), /var/log/secure (auth attempts, privilege escalation), /var/log/audit/audit.log (system call traces if auditd enabled), /var/log/XR/logging (IOS XR-specific event log). Capture 'show users' (active sessions at time of analysis). Export command history if available (bash_history for any user shells spawned). Hash all logs (SHA-256) before analysis.

Step 5, Communication: Notify network operations, change management, and relevant stakeholders of the patching timeline; escalate to leadership if patching critical routing infrastructure requires a maintenance window that affects availability SLAs.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (Preparation: communication and coordination)

Controls: NIST 800-53 IR-4 (Incident Handling), CIS 5.1 (Establish and Maintain a Secure Configuration Management Process)

Compensating: If formal escalation process is absent, create a simple email/ticket template: (1) list affected devices and versions, (2) required maintenance window duration and risk of exploitation, (3) proposed patch date, (4) impact to SLAs or customer service, (5) contingency (rollback plan). Send to VP of Network Ops, Security Lead, and relevant service owners. Document all approvals and rejections in the change ticket. For core infrastructure, recommend a phased approach: patch non-critical devices first, then schedule core routers during planned maintenance.

Evidence: Preserve communication trail: email approvals, change tickets with timelines, downtime notifications to customers (if applicable). Document SLA impact analysis. Keep signed-off change management forms. These form the incident communication record.

Step 6, Long-Term: Review local access provisioning controls for network devices; evaluate whether privileged access management (PAM) tooling or jump-host enforcement reduces exposure surface for similar local-access vulnerabilities in future.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4.4 (Post-Incident Activities: lessons learned)

Controls: NIST 800-53 AC-2 (Account Management), NIST 800-53 AC-3 (Access Enforcement), NIST 800-53 IA-5 (Authentication), CIS 6.1 (Establish a Process for Secure Onboarding)

Compensating: If PAM tool is not budgeted, implement jump-host enforcement manually: (1) disable all direct SSH access to production IOS XR devices, (2) require all CLI access to route through a hardened Linux jump host with centralized auditd/syslog, (3) provision only service accounts on routers (no interactive users), (4) use SSH key-pair auth with restricted SSH configs (command=' restrictions in authorized_keys). Document this in a baseline security hardening guide. Periodically audit jump-host logs for unauthorized command attempts. This reduces local-access attack surface significantly at near-zero cost.

Evidence: Document the long-term control architecture: baseline security policies, jump-host configuration (sshd_config, sudoers rules), audit logging strategy (syslog centralization, log retention). Preserve lessons-learned meeting notes and control design documents. Reference these in annual security reviews.

Detection Guidance

Both vulnerabilities require authenticated local CLI access, so detection focuses on access anomalies and privilege changes rather than network-borne indicators. On IOS XR devices, review AAA (Authentication, Authorization, and Accounting) logs for unexpected local authentication events, particularly outside change windows or from accounts not normally active on the device. Look for CLI command execution logs showing OS-level or shell-escape commands issued by restricted-privilege accounts. If syslog forwarding is configured, query your SIEM for events from IOS XR devices containing privilege-change indicators or unexpected process spawning. Review any task-group or user-role modifications in configuration history. For devices with XR Linux shell access enabled, audit shell access logs for root-level command execution not initiated by expected automation accounts. Note: no confirmed IOCs, exploit code, or behavioral signatures have been publicly released as of the advisory disclosure; detection relies on access and privilege telemetry rather than known malicious patterns.

Framework Mappings

MITRE-ATTACK

- **T1078** — Valid Accounts
- **T1059** — Command and Scripting Interpreter
- **T1068** — Exploitation for Privilege Escalation

NIST-800-53R5

- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation

CIS-V8

- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.23** — Information security for use of cloud services

SOC2-TSC

- **CC6.3** — Authorizes, modifies, or removes access

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1078	Valid Accounts	Defense-Evasion
T1059	Command and Scripting Interpreter	Execution
T1068	Exploitation for Privilege Escalation	Privilege-Escalation

Sources

Source	URL	Tier
Cyberpress	https://cyberpress.org/critical-cisco-ios-xr-vulnerability-allows-a...	T3
Cisco IOS XR Software CLI Privilege Escalation Vulnerability	https://sec.cloudapps.cisco.com/security/center/content/CiscoSecuri...	T3
Cisco IOS XR Software Command Injection Vulnerability	https://www.cisco.com/c/en/us/support/docs/csa/cisco-sa-xr-cmdinj-v...	T3
Cisco Patches High-Severity IOS XR Vulnerabilities	https://www.securityweek.com/cisco-patches-high-severity-ios-xr-vul...	T3
Cisco patched 4 vulnerabilities in it's IOS XR software. The two most critical ...	https://x.com/CCBalert/status/2032142245864804570	T3

Generated 2026-03-29 18:39 UTC by TJS Security Command Center