INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-07-07 15:06 UTC

CVE-2026-8924: Trailing Dot Domain Super Cookie in Microsoft Azure Linux 3.0 curl

CVE VULNERABILITY | CRITICAL | CVSS 9.1

SCC Item ID	SCC-CVE-2026-0394
Type	CVE Vulnerability
CVE ID	CVE-2026-8924
Severity	CRITICAL
CVSS Base Score	9.1
EPSS Score	0.0054 (41th percentile)
Affected Products	Microsoft azl3 curl 8.11.1-9 on Azure Linux 3.0
Published	2026-07-07T01:43:20
Discovery Source	Msrc Patch Tuesday

Executive Summary

A critical vulnerability (CVSS 9.1) in Microsoft's Azure Linux 3.0 packaging of curl allows cookies to be scoped more broadly than intended through trailing dot domain handling, according to MSRC and the upstream curl project advisory. Organizations running Azure Linux 3.0 workloads that use curl for authenticated HTTP communications are affected. If exploited, the flaw could allow an attacker to intercept or inject session cookies across domain boundaries, potentially compromising authenticated sessions in cloud-hosted applications.

Technical Analysis

CVE-2026-8924 affects Microsoft azl3 curl version 8.11.1-9 on Azure Linux 3.0. The vulnerability resides in curl's handling of the trailing dot in cookie domain attributes. When a server sets a cookie with a trailing dot in the domain field (e.g., '.example.com.'), curl incorrectly scopes the cookie, potentially allowing it to be sent to unintended domains or subdomains. This behavior enables cross-domain cookie access or cookie injection (MITRE T1539, Steal Web Session Cookie; T1550.004, Web Session Cookie abuse). Weakness classification: CWE-20 (Improper Input Validation) and CWE-565 (Reliance on Cookies without Validation and Integrity Checking). CVSS base score: 9.1. EPSS score: 0.00536 (41st percentile), indicating low current exploitation probability. The CVE is not listed in CISA KEV as of disclosure. The upstream curl project published a dedicated advisory at curl.se. Disclosed via Microsoft Patch Tuesday July 2026 (MSRC). No public exploit code or active threat actor exploitation is reported in the source material.

Action Checklist

1. Step 1: Containment, Identify all Azure Linux 3.0 systems running curl 8.11.1-9 using your asset inventory (CIS 1.1). Isolate or restrict outbound HTTP/HTTPS traffic from affected hosts to untrusted or cross-domain endpoints where session cookies are in use, until patching is complete (CIS 5.1, CIS 5.2).
2. Step 2: Detection, Query your SIEM for curl user-agent strings originating from Azure Linux 3.0 hosts making requests to multiple distinct domains within a short session window, which may indicate cookie leakage. Review audit log coverage (NIST AU-2, AU-12; CIS 6.2) for HTTP client activity on affected hosts. Check for anomalous Set-Cookie headers containing trailing dot domain attributes in HTTP response logs.
3. Step 3: Eradication, Apply the updated curl package for Azure Linux 3.0 per the MSRC July 2026 Patch Tuesday advisory (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-8924) and the upstream curl advisory at https://curl.se/docs/CVE-2026-8924.html. Confirm the remediated version supersedes 8.11.1-9. Use automated patch management per CIS 7.3 and CIS 7.4.
4. Step 4: Recovery, After patching, verify curl version on all Azure Linux 3.0 hosts. Monitor HTTP session logs for residual anomalous cookie-sharing behavior (NIST AU-6). Rotate any session tokens or API keys that were in active use on affected hosts during the exposure window (NIST IA-4, IA-5).
5. Step 5: Post-Incident, Review cookie handling configuration in applications deployed on Azure Linux 3.0 workloads. Assess whether application-layer controls enforce strict cookie domain scoping independently of the underlying curl library. Document the gap in patch lag for third-party packaged libraries in cloud base images and update your vulnerability management process (CIS 7.1, CIS 7.2) accordingly.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and legal/compliance if evidence of actual cross-domain cookie interception is found (e.g., unauthorized authenticated sessions in application logs attributable to leaked cookies), if affected workloads processed PII, PHI, or PCI-scoped session tokens during the exposure window triggering breach notification assessment, or if the organization lacks the patching capability to remediate within 24 hours given the CVSS 9.1 severity and internet-exposed Azure Linux 3.0 workloads.
Recovery Notes	After verifying the curl package upgrade from 8.11.1-9 to the remediated version on all Azure Linux 3.0 hosts, monitor reverse proxy and application HTTP session logs for a minimum of 72 hours for any residual multi-domain cookie-sharing anomalies that could indicate a separate, application-layer cookie scoping misconfiguration unrelated to the curl library bug. Additionally, confirm that any applications that cache curl's cookie jar files (e.g., using `-c cookiejar.txt`) have had those files purged and regenerated post-patch, as stale cookie jars may retain trailing-dot-scoped cookies written by the vulnerable version.

Forensic Artifacts	HTTP response headers captured via tcpdump or PCAP on affected Azure Linux 3.0 hosts, filtered for Set-Cookie headers containing trailing dot domain attributes (e.g., `domain=.example.com.`) — the primary artifact of CVE-2026-8924 exploitation | curl cookie jar files (default: `~/.local/share/curl/cookies` or application-specified `-c` paths) written by curl 8.11.1-9, which may contain cookies scoped to trailing-dot domains that were improperly accepted and would not be present in a correctly functioning curl installation | Reverse proxy and web server access logs (e.g., `/var/log/nginx/access.log`) on upstream services, filtered for the `curl/8.11.1` User-Agent string making sequential authenticated requests to multiple distinct subdomains within the same session, indicating potential cookie leakage across domain boundaries | Pre-patch curl binary SHA-256 hash and `rpm -qi curl` output from affected Azure Linux 3.0 hosts, establishing forensic proof of the vulnerable package version 8.11.1-9 present during the exposure window | Application environment variable exports (`/proc//environ`) and shell history files on affected hosts, identifying curl invocations that used cookie-related flags (`-b`, `-c`, `-H 'Cookie:'`) against multi-domain targets during the exposure window — scoping which credential material and session tokens were at risk

Per-Action IR Details

Step 1: Containment — Identify all Azure Linux 3.0 systems running curl 8.11.1-9 using your asset inventory (CIS 1.1). Isolate or restrict outbound HTTP/HTTPS traffic from affected hosts to untrusted or cross-domain endpoints where session cookies are in use, until patching is complete.

NIST Phase: Containment

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

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), NIST AC-4 (Information Flow Enforcement), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: Run `rpm -q curl` on all Azure Linux 3.0 hosts via a bash loop or Ansible ad-hoc command (`ansible all -m command -a 'rpm -q curl' -l azl3_hosts`) to identify affected instances returning version 8.11.1-9. Apply host-based outbound firewall rules using `firewall-cmd` or `iptables` to restrict HTTPS egress from affected hosts to only allowlisted internal domains until the patch is applied.

Evidence: Before isolating any host, capture active outbound HTTP/HTTPS connections with `ss -tnp` or `netstat -tnp` and save the output — this preserves evidence of live cross-domain curl sessions that may reflect in-progress cookie leakage. Also capture `/proc/net/tcp` and `/proc/net/tcp6` snapshots and running curl process arguments via `ps auxww | grep curl` to identify which applications and target domains were active at containment time.

Step 2: Detection — Query your SIEM for curl user-agent strings originating from Azure Linux 3.0 hosts making requests to multiple distinct domains within a short session window, which may indicate cookie leakage. Review AU-2 and AU-12 audit log coverage for HTTP client activity on affected hosts. Check for anomalous Set-Cookie headers containing trailing dot domain attributes in HTTP response logs.

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), CIS 8.2 (Collect Audit Logs)

Compensating: On affected Azure Linux 3.0 hosts, enable curl verbose logging by setting `CURLOPT_VERBOSE` or invoking curl with `-v` redirected to a log file, then grep for `Set-Cookie` response headers containing a trailing dot in the domain attribute (e.g., `grep -i 'set-cookie.*domain=\.' /var/log/app/curl_verbose.log`). Use tcpdump to capture and inspect HTTP response headers in real time: `tcpdump -A -s 0 'tcp port 80 or tcp port 443' | grep -i 'set-cookie'`. For TLS-encrypted sessions, capture PCAP and decrypt using application-exported session keys from `SSLKEYLOGFILE` if configurable.

Evidence: Capture full HTTP response headers from network traffic before any session termination or credential rotation — trailing dot domain Set-Cookie headers (e.g., `domain=.example.com.`) are transient and only present in

live or recently cached traffic. Collect `/var/log/nginx/access.log`, `/var/log/apache2/access.log`, or equivalent reverse proxy logs on any upstream servers receiving requests from affected hosts, filtering for the curl/8.11.1 user-agent string making sequential requests to multiple distinct domain names within the same session window.

Step 3: Eradication — Apply the updated curl package for Azure Linux 3.0 per the MSRC July 2026 Patch Tuesday advisory (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-8924) and the upstream curl advisory at https://curl.se/docs/CVE-2026-8924.html. Confirm the remediated version supersedes 8.11.1-9. Use automated patch management per CIS 7.3 and CIS 7.4.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 7.2 (Establish and Maintain a Remediation Process), NIST SI-2 (Flaw Remediation)

Compensating: Apply the patch manually via `tdnf update curl` on each Azure Linux 3.0 host (tdnf is the native package manager for Azure Linux). After update, verify the installed version with `rpm -q curl` and confirm it is no longer 8.11.1-9. If automated patching is unavailable, script the update and version check across all affected hosts using SSH in a bash loop: `for host in $(cat azl3_hosts.txt); do ssh $host 'tdnf update -y curl && rpm -q curl'; done`.

Evidence: Before applying the patch, snapshot the current curl binary and shared library state: record `rpm -qi curl` output and capture `sha256sum /usr/bin/curl` and `ldd /usr/bin/curl` to establish a pre-patch baseline. These preserve evidence of the vulnerable binary version for post-incident documentation and confirm that the patched binary genuinely replaces the affected one. Capture any application restart or service reload events from systemd journals (`journalctl -u --since 'patch_time'`) to confirm affected applications reloaded the updated curl library.

Step 4: Recovery — After patching, verify curl version on all Azure Linux 3.0 hosts. Monitor HTTP session logs for residual anomalous cookie-sharing behavior (AU-6). Rotate any session tokens or API keys that were in active use on affected hosts during the exposure window (D3-CRO — Credential Rotation).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

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

Compensating: Run `rpm -q curl` across all Azure Linux 3.0 hosts to confirm the vulnerable version 8.11.1-9 is absent. For credential rotation without an enterprise secrets manager, enumerate active API keys and session tokens using application-specific configuration files (e.g., `~/.netrc`, environment variables exported to curl via scripts) on affected hosts, revoke them via the issuing service's API or admin console, and reissue. Monitor residual cookie anomalies for 72 hours post-patch by reviewing reverse proxy access logs for the curl/8.11.1 user-agent making multi-domain requests — this user-agent should disappear after patching.

Evidence: Before rotating credentials, capture a list of all session tokens and API keys in use on affected hosts during the exposure window by reviewing application configuration files, environment variable exports (`/proc//environ` for running processes), and shell history files (`~/.bash_history`) for curl invocations that included `-b` (cookie file), `-c` (cookie jar), or `-H 'Cookie:'` flags targeting multiple distinct domains — these identify which credential material was potentially exposed via cross-domain cookie leakage.

Step 5: Post-Incident — Review cookie handling configuration in applications deployed on Azure Linux 3.0 workloads. Assess whether application-layer controls enforce strict cookie domain scoping independently of the underlying curl library. Document the gap in patch lag for third-party packaged libraries in cloud base images and update your vulnerability management process (CIS 7.1, CIS 7.2) accordingly.

NIST Phase: Post Incident

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

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.2 (Ensure Authorized Software is Currently Supported), NIST AU-11 (Audit Record

Retention)

Compensating: Audit all applications on Azure Linux 3.0 workloads that invoke curl programmatically or via libcurl, checking whether they set `CURLOPT_COOKIEDOMAIN` with strict domain scoping or use `SameSite=Strict` and `Domain` attributes explicitly in their own cookie-setting logic. Document findings in a patch lag report that captures the delta between upstream curl advisory publication and Microsoft azl3 package availability, and use this to calibrate SLA thresholds in your vulnerability management process for cloud-vendor-repackaged open-source libraries.

Evidence: Retain all collected artifacts for the post-incident review period per your data retention policy (NIST AU-11): pre-patch curl binary hashes, PCAP captures of HTTP sessions showing Set-Cookie header anomalies, curl verbose logs, and asset inventory exports of affected Azure Linux 3.0 hosts. These form the evidentiary record for determining the exposure window, scope of potentially compromised session tokens, and whether any cross-domain cookie leakage resulted in unauthorized authenticated access to downstream services.

Detection Guidance

Query HTTP client logs on Azure Linux 3.0 hosts for curl requests where response Set-Cookie headers include a domain attribute ending in a trailing dot (e.g., 'domain=example.com.'). Look for session cookies being transmitted to domains outside the expected first-party scope for a given application. In SIEM, correlate curl-originated requests from a single host to more than one distinct registered domain within the same session timeframe, this may indicate cookie leakage across domain boundaries (aligned with MITRE T1539). Enable and review audit logging for HTTP client activity; ensure audit record generation is active on affected hosts (NIST AU-2, AU-12). No public IOCs (IPs, hashes, domains) are associated with active exploitation in the source material. Detection at this stage is primarily configuration and version-based: confirm curl version 8.11.1-9 presence via package manager queries (e.g., 'rpm -q curl' on Azure Linux 3.0 hosts).

Framework Mappings

MITRE-ATTACK

- **T1539** — Steal Web Session Cookie
- **T1550.004** — Web Session Cookie

OWASP-TOP10-2021

- **A03:2021** — Injection

NIST-800-53R5

- **SI-10** — Information Input Validation

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

ISO-27001-2022

- **A.8.26** — Application security requirements
- **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
T1539	Steal Web Session Cookie	Credential-Access
T1550.004	Web Session Cookie	Defense-Evasion

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-8924	T1
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Jul	T1
CVE-2026-8924 - Vulnerability Details - OpenCVE	https://app.opencve.io/cve/CVE-2026-8924	T3
CVE-2026-8924 - trailing dot domain super cookie	https://cvefeed.io/vuln/detail/CVE-2026-8924	T3
curl - trailing dot domain super cookie - CVE-2026-8924	https://curl.se/docs/CVE-2026-8924.html	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-8924	T1

Generated 2026-07-07 15:06 UTC by TJS Security Command Center