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

# CVE-2026-8926: Password Leak via netrc and User-in-URL in Microsoft Azure Linux curl

CVE VULNERABILITY | CRITICAL | CVSS 9.1

| | |
|---|---|
| SCC Item ID | SCC-CVE-2026-0395 |
| Type | CVE Vulnerability |
| CVE ID | CVE-2026-8926 |
| Severity | CRITICAL |
| CVSS Base Score | 9.1 |
| EPSS Score | 0.0048 (38th 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

Microsoft disclosed a critical credential exposure vulnerability (CVE-2026-8926) in its Azure Linux 3.0-packaged curl utility (azl3 curl 8.11.1-9), as part of the July 2026 Patch Tuesday release. When a URL contains embedded user credentials and a .netrc file is present, the affected curl build may leak passwords, according to MSRC. Organizations running workloads on Azure Linux 3.0 that use curl for authenticated transfers, including automated scripts, CI/CD pipelines, and service integrations, should treat this as a credential exposure risk requiring prompt patching.

## Technical Analysis

CVE-2026-8926 affects Microsoft's Azure Linux 3.0 packaging of curl, specifically azl3 curl 8.11.1-9. The vulnerability arises from an interaction between URL-embedded credentials (user:password@host format) and .netrc file parsing: under specific conditions, curl may disclose the password from one of these sources inappropriately. This falls into a documented class of curl credential-handling flaws, the same class as upstream CVEs CVE-2023-27776 and CVE-2024-11053, but this identifier is specific to Microsoft's Azure Linux package, not upstream curl. CVSS base score is 9.1 (Critical); EPSS score is 0.479% (38th percentile), indicating low observed exploitation probability at time of publication. CWEs: CWE-200 (Exposure of Sensitive Information to an Unauthorized Actor) and CWE-522 (Insufficiently Protected Credentials). MITRE ATT&CK techniques: T1552 (Unsecured Credentials) and T1552.001 (Credentials In Files). No CISA KEV listing as of analysis date. Source note: The primary source is MSRC (T1 tier). NVD and CVE.org population status for CVE-2026-8926 was not

confirmed as of analysis date. CVSS vector string is pending NVD publication.

## Action Checklist

1. Step 1: Containment, Identify all Azure Linux 3.0 systems running azl3 curl 8.11.1-9. Prioritize systems where curl is invoked with URL-embedded credentials or where .netrc files exist in user home directories or service account paths. Restrict or temporarily disable automated curl processes using both credential sources until patching is complete. Source: MSRC Update Guide for CVE-2026-8926.
2. Step 2: Detection, Audit .netrc files across Azure Linux 3.0 hosts using 'find / -name .netrc 2>/dev/null' and inventory scripts that pass credentials via URL (grep for patterns matching 'user:pass@' in cron jobs, CI/CD configs, and shell scripts). Review NIST SP 800-53 Rev. 5 AU-2 (Event Logging) and AU-12 (Audit Record Generation) sources, specifically shell history, sudo logs, and any API gateway or proxy access logs, for curl invocations combining both credential methods. Check for outbound connections to unexpected hosts immediately after such invocations (NIST AU-6).
3. Step 3: Eradication, Apply the Microsoft July 2026 Patch Tuesday update for Azure Linux 3.0 that addresses CVE-2026-8926. Verify the updated package version is installed via 'tdnf info curl' (Azure Linux 3.0 package manager) or equivalent for your distribution. After patching, audit and remediate all scripts using URL-embedded credentials: migrate to .netrc-only or environment-variable credential passing, not both simultaneously. Remove or restrict .netrc files to the minimum required service accounts (NIST AC-6, CIS 5.4).
4. Step 4: Recovery, After patching, rotate all credentials that may have been passed via curl on affected systems, particularly any credentials present in both URL-embedded and .netrc formats (D3-CRO: Credential Rotation). Validate patch application with a package version check. Monitor authentication logs for the rotated accounts for 30 days for anomalous access patterns. Re-enable automated curl processes only after credential rotation is confirmed (NIST SP 800-53 Rev. 5 IR-4, AU-6).
5. Step 5: Post-Incident, Review your inventory process to confirm Azure Linux 3.0 package-level vulnerabilities are captured in your vulnerability management workflow (CIS 7.1, CIS 7.2). Assess whether credential-in-URL patterns are used elsewhere in your environment beyond curl and document findings. Implement a policy prohibiting URL-embedded credentials in scripts and enforce .netrc file permissions (mode 600, owned by the invoking user) as a standing configuration standard (NIST AC-6, CIS 4.6, D3-CH: Credential Hardening).

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | IMMEDIATE |
| **Escalation Criteria** | Escalate immediately to CISO and legal/privacy counsel if forensic analysis of proxy logs, DNS query logs, or authentication logs reveals that credentials stored in .netrc files or passed via URL to azl3 curl 8.11.1-9 were transmitted to any destination other than the intended upstream service, as this constitutes confirmed credential exfiltration and may trigger breach notification obligations under applicable data protection regulations. |

| | |
|---|---|
| **Recovery Notes** | Re-enable automated curl processes only after confirming the patched curl version is installed ('tdnf info curl' no longer reports azl3 8.11.1-9) and all credentials identified in .netrc files and URL-embedded invocations have been rotated at the upstream service. Monitor authentication logs (/var/log/auth.log, /var/log/secure, and upstream service access logs) for 30 days post-rotation for any authentication attempt using pre-rotation credential values, which would indicate an attacker captured and is replaying the leaked credentials. Document the full exposure window — from the earliest recorded invocation of the vulnerable curl build with both credential sources present to the confirmed patch date — for regulatory and insurance reporting purposes. |
| **Forensic Artifacts** | /proc//cmdline for running curl processes: exposes URL-embedded credentials (user:pass@host) in cleartext in kernel memory for any curl invocation active at time of discovery on azl3 curl 8.11.1-9 hosts \| /var/log/audit/audit.log EXECVE records: captures the full command line of each curl invocation including both the '-n'/'--netrc' flag and any 'user:pass@' URL argument, establishing a timeline of credential-combined invocations specific to CVE-2026-8926's trigger condition \| All .netrc files located via 'find / -name .netrc': contain plaintext credentials for services authenticated by curl on affected Azure Linux 3.0 hosts — these define the complete set of secrets requiring rotation and document the scope of potential leakage \| Outbound proxy or DNS logs correlated to curl process start times: an attacker receiving credentials leaked by CVE-2026-8926 would appear as an anomalous destination host in connection records timestamped within seconds of the vulnerable curl invocation — the specific exploit mechanism makes the destination-to-invocation time correlation forensically significant \| CI/CD pipeline job logs (GitHub Actions, Jenkins, Azure DevOps runner output): capture curl invocations with URL-embedded credentials as they appear in pipeline stdout/stderr, which may persist in build artifact storage long after the host-level evidence is gone — these logs document exposure in automated pipeline contexts specifically called out in the CVE advisory |

### Per-Action IR Details

**Step 1: Containment — Identify all Azure Linux 3.0 systems running azl3 curl 8.11.1-9. Prioritize systems where curl is invoked with URL-embedded credentials or where .netrc files exist in user home directories or service account paths. Restrict or temporarily disable automated curl processes using both credential sources until patching is complete. Source: MSRC Update Guide for CVE-2026-8926.**

**NIST Phase:** Containment

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

**Controls:** NIST AC-2 (Account Management), NIST AC-6 (Least Privilege), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

**Compensating:** Run 'tdnf info curl | grep -E "Version|Release"' on each Azure Linux 3.0 host to confirm azl3 curl 8.11.1-9; automate across fleet with a bash loop over SSH: 'for h in $(cat hosts.txt); do ssh $h "tdnf info curl | grep -E Version\|Release"; done'. Use 'find / -name .netrc 2>/dev/null' and 'grep -rE "[a-zA-Z0-9+_.-]+:[^@]+@" /etc/cron* /var/spool/cron /opt /home 2>/dev/null' to surface both credential patterns. Temporarily comment out or 'chmod 000' automated curl wrapper scripts identified as using both credential sources.

**Evidence:** Before disabling any automated curl process, capture: (1) output of 'ss -tnp' or 'netstat -ano' to record all active outbound connections from the process at the moment of discovery — a credential leak may have already initiated a session to an unexpected host; (2) running process list ('ps auxf') showing curl invocations with command-line arguments, which may expose URL-embedded credentials in cleartext in '/proc//cmdline'; (3) contents of any in-flight /tmp files or named pipes used by the curl process. These are gone the moment the process is killed or the host is rebooted.

**Step 2: Detection — Audit .netrc files across Azure Linux 3.0 hosts using 'find / -name .netrc 2>/dev/null' and inventory scripts that pass credentials via URL (grep for patterns matching 'user:pass@' in cron jobs, CI/CD configs, and shell scripts). Review AU-2 (Event Logging) and AU-12 (Audit Record Generation) sources — specifically shell history, sudo logs, and any API gateway or proxy access logs — for curl invocations combining both credential methods. Check for outbound connections to unexpected hosts immediately after such invocations (NIST AU-6).**

**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 AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

**Compensating:** Deploy Sysmon on Azure Linux 3.0 hosts (via the Linux Sysmon build) with an EventID 3 (NetworkConnect) rule filtering on processes named 'curl' to capture destination IP, port, and timestamp of each outbound connection. Parse /var/log/audit/audit.log with 'ausearch -c curl' to retrieve execve syscall records showing full curl command lines including URL-embedded credentials. Review /root/.bash_history and /home/*/.bash_history and service account shell histories under /var/lib// for 'curl.*:.*@' patterns. For CI/CD, pull runner job logs and search for curl invocations with both '--netrc' or '-n' flags and a URL containing '@'.

**Evidence:** Capture before any log rotation or history truncation: (1) full contents of all .netrc files found (record permissions, owner, and plaintext credentials present — these confirm what secrets were at risk); (2) /proc//cmdline for any currently running curl processes, which exposes URL-embedded credentials in memory; (3) /var/log/audit/audit.log entries for EXECVE records of curl with both '-n'/'--netrc' and a user:pass@ URL argument; (4) outbound DNS query logs or proxy access logs showing destination hostnames contacted by the vulnerable curl invocations — an attacker receiving leaked credentials would appear as an anomalous destination here; (5) timestamps and source IPs from any API gateway logs showing authenticated requests from Azure Linux 3.0 hosts using credentials that match those found in .netrc files.

**Step 3: Eradication — Apply the Microsoft July 2026 Patch Tuesday update for Azure Linux 3.0 that addresses CVE-2026-8926. Verify the updated package version is installed via 'tdnf info curl' or equivalent. After patching, audit and remediate all scripts using URL-embedded credentials: migrate to .netrc-only or environment-variable credential passing, not both simultaneously. Remove or restrict .netrc files to the minimum required service accounts (NIST AC-6, CIS 5.4).**

**NIST Phase:** Eradication

**Reference:** NIST 800-61r3 §3.4 — Eradication

**Controls:** NIST AC-6 (Least Privilege), NIST SI-2 (Flaw Remediation), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

**Compensating:** Apply the patch via 'tdnf update curl' on each Azure Linux 3.0 host; confirm remediation with 'tdnf info curl' and validate the package version no longer matches azl3 curl 8.11.1-9. For script remediation, use 'grep -rE "curl .* [a-zA-Z0-9+_.-]+:[^@]+@" /etc /opt /home /var/spool/cron 2>/dev/null' to enumerate all remaining URL-embedded credential usages. Enforce .netrc permissions with 'find / -name .netrc -exec chmod 600 {} \; -exec chown {} \;' and remove .netrc files for service accounts that no longer require them.

**Evidence:** Before applying the patch, capture: (1) the pre-patch package manifest ('tdnf list installed | grep curl') as forensic evidence of the vulnerable version's presence and patch window duration; (2) a full file listing and hash (sha256sum) of all .netrc files that will be removed or modified — preserve copies in a secured forensic store, as their contents represent the credential set requiring rotation; (3) a snapshot of all cron jobs and CI/CD pipeline configurations containing 'user:pass@' URL patterns before they are remediated, to document the full scope of exposure for the post-incident report.

**Step 4: Recovery — After patching, rotate all credentials that may have been passed via curl on affected systems, particularly any credentials present in both URL-embedded and .netrc formats (D3-CRO: Credential Rotation). Validate patch application with a package version check. Monitor authentication logs for the rotated**

**accounts for 30 days for anomalous access patterns. Re-enable automated curl processes only after credential rotation is confirmed (NIST IR controls, AU-6).**

**NIST Phase:** Recovery

**Reference:** NIST 800-61r3 §3.5 — Recovery

**Controls:** NIST AC-2 (Account Management), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 5.2 (Use Unique Passwords), CIS 6.2 (Establish an Access Revoking Process)

**Compensating:** For each credential set identified in Step 2 as exposed (URL-embedded and/or .netrc), rotate the secret at the upstream service (API keys via the provider portal, service account passwords via 'passwd ' or Azure AD/Entra ID reset). Update all downstream consumers — CI/CD pipeline secrets, vault references, and .netrc files — to the new credential before re-enabling curl automation. Monitor /var/log/auth.log and /var/log/secure on each previously affected host for authentication events tied to the old credential values (pre-rotation) for 30 days; any post-rotation authentication attempt using the old credential is a strong indicator of active compromise and requires immediate escalation.

**Evidence:** Before re-enabling automated curl processes, confirm: (1) 'tdnf info curl' shows the patched version replacing azl3 curl 8.11.1-9; (2) all .netrc files have been updated to new credentials and permissions are mode 600; (3) no process in 'ps auxf' output shows a curl invocation still referencing old 'user:pass@host' URL patterns. Retain the forensic record of old credential values (hashed/obscured) to enable detection if those values appear in authentication logs during the 30-day monitoring window.

**Step 5: Post-Incident — Review your inventory process to confirm Azure Linux 3.0 package-level vulnerabilities are captured in your vulnerability management workflow (CIS 7.1, CIS 7.2). Assess whether credential-in-URL patterns are used elsewhere in your environment beyond curl and document findings. Implement a policy prohibiting URL-embedded credentials in scripts and enforce .netrc file permissions (mode 600, owned by the invoking user) as a standing configuration standard (NIST AC-6, CIS 4.6, D3-CH: Credential Hardening).**

**NIST Phase:** Post Incident

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

**Controls:** NIST AC-6 (Least Privilege), CIS 4.6 (Securely Manage Enterprise Assets and Software), 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)

**Compensating:** Add an osquery scheduled query targeting Azure Linux 3.0 hosts to detect .netrc files with permissions other than 600 or owned by non-invoking users: 'SELECT path, username, permissions FROM file JOIN users ON file.uid = users.uid WHERE filename = ".netrc"'. Author a Sigma rule detecting shell process execution where CommandLine matches 'curl' AND contains the regex '[a-zA-Z0-9._%+-]+:[^@]+@' to catch URL-embedded credential usage in future script deployments. Add azl3 curl to the monitored package list in your vulnerability scanner (OpenVAS or Trivy for container images) with an alert threshold for any version matching 8.11.1-9.

**Evidence:** Document for the lessons-learned record: (1) the complete list of Azure Linux 3.0 hosts that had azl3 curl 8.11.1-9 installed and the duration of exposure (install date to patch date, recoverable from 'rpm -qi curl --queryformat "%{INSTALLTIME}"'); (2) the full inventory of .netrc files and URL-embedded credential patterns discovered, with owning service accounts and the upstream services they authenticate to — this is the authoritative blast-radius document; (3) the patch-to-detection gap (time between Microsoft's July 2026 Patch Tuesday disclosure and your first confirmed identification of the vulnerable package in your environment), which drives the remediation SLA update in your vulnerability management policy.

## Detection Guidance

On Azure Linux 3.0 hosts, query installed package version: run 'tdnf info curl' and flag any host returning azl3 curl 8.11.1-9. Identify .netrc files: 'find /home /root /var /etc -name .netrc 2>/dev/null'. Search for curl invocations with embedded credentials in scripts, cron jobs, and CI/CD pipeline definitions: grep recursively for patterns

matching 'https?://[^:]+:[^@]+@' in /etc/cron*, /var/spool/cron, ~/.bashrc, and pipeline config files. In SIEM, correlate curl process execution events (audit logs, auditd records for execve syscalls invoking curl) where both a .netrc file exists in the user's home directory and a URL argument matches the credential-embedded pattern. No public IOCs (IPs, domains, hashes) are associated with active exploitation of this CVE at the time of analysis. The EPSS score of 0.479% (38th percentile) indicates low observed exploitation probability, and the absence from CISA KEV supports this assessment. Detection priority should focus on configuration exposure, not active exploitation indicators. Reference NIST SP 800-53 Rev. 5 AU-2 and AU-12 for log source requirements. D3-LAM (Local Account Monitoring) and D3-SFA (System File Analysis) apply for .netrc file auditing.

## Framework Mappings

### MITRE-ATTACK

- **T1552.001** — Credentials In Files
- **T1552** — Unsecured Credentials

### OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A04:2021** — Insecure Design
- **A07:2021** — Identification and Authentication Failures

### NIST-800-53R5

- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest
- **IA-5** — Authenticator Management

### HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.308(a)(5)(ii)(D)** — Password Management
- **164.312(d)** — Person or Entity Authentication

### CIS-V8

- **5.2** — Use Unique Passwords
- **6.3** — Require MFA for Externally-Exposed Applications
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

### SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

### 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 |
|---|---|---|
| T1552.001 | Credentials In Files | Credential-Access |
| T1552 | Unsecured Credentials | Credential-Access |

## Sources

| Source | URL | Tier |
|---|---|---|
| **MSRC Update Guide** | https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-8926 | T1 |
| **(consolidated)** | https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Jul | T1 |
| **CVE-2026-8926 - Vulnerability Details - OpenCVE** | https://app.opencve.io/cve/CVE-2026-8926 | T3 |
| **CVE-2026-24926 Detail - NVD** | https://nvd.nist.gov/vuln/detail/CVE-2026-24926 | T1 |
| **CVE-2026-8926 - CVE Record** | https://www.cve.org/CVERecord?id=CVE-2026-8926 | T1 |
| **NVD** | https://nvd.nist.gov/vuln/detail/CVE-2026-8926 | T1 |



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