INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-29 06:52 UTC

CVE-2026-31657: Critical batman-adv Kernel Use-After-Free in Azure Linux 3.0

CVE VULNERABILITY | CRITICAL | CVSS 9.8

SCC Item ID	SCC-CVE-2026-0088
Type	CVE Vulnerability
CVE ID	CVE-2026-31657
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.0006 (18th percentile)
Affected Products	Microsoft azl3 kernel 6.6.130.1-3 on Azure Linux 3.0
Published	2026-04-29T01:47:49
Discovery Source	Msrc Patch Tuesday

Executive Summary

A critical use-after-free vulnerability (CVE-2026-31657, CVSS 9.8) in the batman-adv kernel module affects Microsoft's Azure Linux 3.0 kernel package (azl3 version 6.6.130.1-3). Organizations running workloads on Azure Linux 3.0 are at risk of memory corruption, privilege escalation, or remote code execution if the vulnerable kernel version remains unpatched. Microsoft addressed this in the April 2026 Patch Tuesday release; immediate patch deployment is the required action.

Technical Analysis

CVE-2026-31657 is a use-after-free (CWE-416) and improper reference counting (CWE-911) vulnerability in the batman-adv (Better Approach To Mobile Adhoc Networking Advanced) kernel module. The defect lies in the failure to hold backbone gateway objects by reference during claim processing, leaving dangling pointers that can be exploited to achieve memory corruption. The CVSS 9.8 base score reflects network-exploitable attack vector, low attack complexity, no privileges required, and no user interaction, characteristics consistent with MITRE ATT&CK T1068 (Exploitation for Privilege Escalation). Affected package: Microsoft azl3 kernel 6.6.130.1-3 on Azure Linux 3.0. The vulnerability was patched in Microsoft's April 2026 Patch Tuesday release. EPSS score is currently low (0.057%, 17.6th percentile) and it is not listed on CISA KEV, indicating no confirmed active exploitation at time of publication. CVSS vector string not available in source materials reviewed at publication time; verify against NVD or MSRC for complete scoring details. Sources: MSRC Update Guide (CVE-2026-31657), NVD (CVE-2026-31657).

Action Checklist

1. Step 1: Containment, Identify all Azure Linux 3.0 systems running azl3 kernel version 6.6.130.1-3. Audit whether batman-adv is loaded ('lsmod | grep batman_adv'). If the module is loaded and the system is internet-facing or multi-tenant, consider restricting network exposure or disabling batman-adv if mesh networking is not required ('rmmod batman_adv' and blacklist the module in /etc/modprobe.d/).
2. Step 2: Detection, Determine affected scope by inventorying kernel versions across Azure Linux 3.0 hosts ('uname -r' or MDVM/Defender for Cloud asset query). Check whether batman-adv is loaded or set to auto-load. Review dmesg and kernel logs for memory corruption indicators (null pointer dereferences, kernel panics, BUG: unable to handle kernel NULL pointer). Correlate with Defender for Cloud CVE findings filtered on CVE-2026-31657.
3. Step 3: Eradication, Apply the patched azl3 kernel package released in Microsoft's April 2026 Patch Tuesday update via the Azure Linux package manager ('tdnf update kernel' or 'dnf update kernel' depending on toolchain). Confirm updated kernel version resolves beyond 6.6.130.1-3 per MSRC advisory. Verify against MSRC Update Guide at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31657.
4. Step 4: Recovery, Reboot affected systems to activate the patched kernel. Confirm running kernel version post-reboot. Validate batman-adv module behavior if mesh networking is required. Run a post-patch vulnerability scan (Tenable, Defender for Cloud, or equivalent) to confirm CVE-2026-31657 is no longer flagged. Monitor system logs for 24-48 hours for anomalous kernel activity.
5. Step 5: Post-Incident, Review whether batman-adv is operationally necessary on affected hosts; disable and blacklist if not in use to reduce kernel attack surface. Evaluate kernel module allowlisting policies. Assess whether Patch Tuesday cadence and Azure Linux update pipelines need tightening to reduce window between patch release and deployment. Map control gap to CIS Benchmark Level 1 for Linux (kernel module restriction controls).

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to incident command if any Azure Linux 3.0 host shows KASAN use-after-free kernel messages in dmesg referencing batman-adv, unexpected privilege escalation events (sudo/su activity from non-admin accounts), or anomalous process spawning from kernel worker threads — any of these indicate active exploitation of CVE-2026-31657 rather than passive vulnerability exposure, triggering full NIST 800-61r3 §3.3 containment and potential breach notification assessment if the host processes regulated data.
Recovery Notes	After rebooting into the patched azl3 kernel, confirm `uname -r` returns a version beyond `6.6.130.1-3` on every previously affected host before returning any system to production traffic. If batman-adv must remain operational post-patch, monitor `journalctl -k` and `dmesg` specifically for KASAN reports or slab corruption errors referencing `batman_adv` for a minimum of 48 hours, as a CVSS 9.8 use-after-free may have introduced pre-exploitation memory corruption that persists until a clean reboot. Validate the recovery against a post-patch Defender for Cloud or Tenable scan confirming CVE-2026-31657 is no longer flagged before closing the incident ticket.

Forensic Artifacts	dmesg / journalctl -k output: Primary forensic source for CVE-2026-31657 exploitation evidence — look specifically for KASAN use-after-free reports, slab corruption messages, or kernel BUG/panic entries that reference batman_adv memory structures, as these are the direct output of the UAF vulnerability being triggered. | /proc/modules and lsmod snapshot: Documents whether batman_adv was loaded at the time of assessment, its dependency chain, and whether it was actively in use — critical for establishing exploitability on each specific host. | rpm -qa kernel output (pre- and post-patch): Provides package-level evidence of the vulnerable azl3 version 6.6.130.1-3 being present, and its replacement by the April 2026 Patch Tuesday fixed package — serves as the primary remediation audit artifact. | /etc/modules-load.d/ and /etc/modprobe.d/ directory contents: Establishes whether batman-adv was configured for automatic loading at boot, determining whether the exposure was persistent across reboots and not just a transient operator action. | Azure Linux audit log (/var/log/audit/audit.log) filtered for SYSCALL records involving privilege escalation (execve of su/sudo, setuid calls) in the timeframe batman-adv was loaded: If CVE-2026-31657 was exploited for local privilege escalation, this log source would capture the resulting privilege change events that follow a successful UAF exploitation chain.

Per-Action IR Details

Step 1: Containment — Identify all Azure Linux 3.0 systems running azl3 kernel version 6.6.130.1-3. Audit whether batman-adv is loaded ('lsmod | grep batman_adv'). If the module is loaded and the system is internet-facing or multi-tenant, consider restricting network exposure or disabling batman-adv if mesh networking is not required ('rmmod batman_adv' and blacklist the module in /etc/modprobe.d/).

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST CM-7 (Least Functionality), NIST SI-4 (System Monitoring), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: On hosts without Defender for Cloud, enumerate all Azure Linux 3.0 nodes via a bash one-liner pushed over SSH: `for host in $(cat hosts.txt); do ssh $host 'uname -r && lsmod | grep batman_adv'; done`. To blacklist immediately without reboot: `echo 'blacklist batman_adv' >> /etc/modprobe.d/batman-blacklist.conf && rmmod batman_adv 2>/dev/null`. Confirm removal with `lsmod | grep batman_adv` returning empty. Use osquery (`SELECT name, used_by FROM kernel_modules WHERE name='batman_adv';`) for fleet-wide module audit if osquery is deployed.

Evidence: Before unloading the module, capture: (1) current module state with `lsmod > /tmp/lsmod_snapshot_$(hostname)_$(date +%s).txt`; (2) any existing kernel ring buffer messages referencing batman-adv or memory corruption with `dmesg | grep -iE 'batman|use-after-free|kernel NULL pointer|BUG:|KASAN|slab corruption' > /tmp/dmesg_pre_containment.txt`; (3) network interface state associated with batman-adv (`ip link show type batadv 2>/dev/null`) to document whether the mesh interface was actively in use and potentially exposed.

Step 2: Detection — Determine affected scope by inventorying kernel versions across Azure Linux 3.0 hosts ('uname -r' or MDVM/Defender for Cloud asset query). Check whether batman-adv is loaded or set to auto-load. Review dmesg and kernel logs for memory corruption indicators (null pointer dereferences, kernel panics, BUG: unable to handle kernel NULL pointer). Correlate with Defender for Cloud CVE findings filtered on CVE-2026-31657.

NIST Phase: Detection Analysis

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

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST RA-5 (Vulnerability Monitoring and Scanning), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 8.2 (Collect Audit Logs)

Compensating: Without MDVM, query kernel version fleet-wide: `pdsh -w ^hosts.txt 'uname -r' | grep '6.6.130.1-3'`. Check for batman-adv auto-load configuration: `grep -r 'batman_adv' /etc/modules /etc/modules-load.d/ /etc/modprobe.d/` on each host. For memory corruption indicators without a SIEM, use: `journalctl -k --since '7 days ago' | grep -iE 'BUG:|KASAN:|use-after-free|batman' > /tmp/kernel_anomaly_review.txt`. Use osquery to check module auto-load config: `SELECT * FROM kernel_modules WHERE name='batman_adv';`.

Evidence: Collect: (1) `/var/log/kern.log` or `journalctl -k` output filtered for KASAN (Kernel Address Sanitizer) reports, slab corruption messages, or `BUG: KASAN: use-after-free` strings — these are the direct fingerprints of CVE-2026-31657 exploitation in batman-adv memory handling; (2) `/proc/modules` snapshot to confirm batman_adv load state and dependency chain; (3) `/etc/modules-load.d/` and `/etc/modprobe.d/` directory listings to determine if auto-load was configured; (4) Defender for Cloud CVE assessment export filtered on `CVE-2026-31657` to establish affected asset count before patching obscures scope.

Step 3: Eradication — Apply the patched azl3 kernel package released in Microsoft's April 2026 Patch Tuesday update via the Azure Linux package manager ('tdnf update kernel' or 'dnf update kernel' depending on toolchain). Confirm updated kernel version resolves beyond 6.6.130.1-3 per MSRC advisory. Verify against MSRC Update Guide at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31657.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-3 (Configuration Change Control), NIST SA-10 (Developer Configuration Management), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: For teams without automated patch orchestration, script sequential patching: `for host in $(cat affected_hosts.txt); do ssh $host 'tdnf update kernel -y && rpm -q kernel'; done`. Verify the patched package version post-update with `rpm -q kernel --queryformat '%{VERSION}-%{RELEASE}\n'` and confirm the installed version exceeds `6.6.130.1-3` per the MSRC April 2026 advisory. Log the pre- and post-patch kernel version for each host to a central CSV for audit evidence: `echo "$host,$(uname -r)" >> patch_audit.csv`. Note: The MSRC URL provided in the original step should be independently verified by the responder against the live MSRC Update Guide, as URL validity cannot be confirmed from this session.

Evidence: Before patching, preserve: (1) `rpm -qa kernel*` output to document the vulnerable azl3 package version `6.6.130.1-3` as the pre-patch baseline; (2) a full `dmesg` dump to capture any pre-patch kernel memory corruption events attributable to batman-adv UAF activity that may indicate prior exploitation; (3) `/proc/version` and `/proc/cmdline` to record exact kernel boot parameters in use on the vulnerable system. This pre-patch forensic snapshot establishes the chain of custody for the vulnerable state.

Step 4: Recovery — Reboot affected systems to activate the patched kernel. Confirm running kernel version post-reboot. Validate batman-adv module behavior if mesh networking is required. Run a post-patch vulnerability scan (Tenable, Defender for Cloud, or equivalent) to confirm CVE-2026-31657 is no longer flagged. Monitor system logs for 24-48 hours for anomalous kernel activity.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

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

Compensating: Post-reboot, verify the patched kernel is active: `uname -r` must return a version beyond `6.6.130.1-3`. If batman-adv must remain loaded, verify the module loaded against the patched kernel using `modinfo batman_adv | grep -E 'version|filename'` and confirm the module path references the new kernel tree. Without Tenable/Defender, run a local OpenSCAP scan using the Azure Linux OVAL feed if available, or manually confirm the fixed package is installed: `rpm -q kernel | grep -v '6.6.130.1-3'`. For 24-48 hour monitoring without a SIEM, schedule: `watch -n 300 'journalctl -k --since -10m | grep -iE "BUG:|KASAN:|batman|panic"'` and redirect output to a rotating log file.

Evidence: Post-reboot, capture: (1) `uname -r` output as the verified remediation record confirming the vulnerable kernel `6.6.130.1-3` is no longer running; (2) `dmesg` from the first clean boot on the patched kernel to establish a

post-remediation kernel health baseline, specifically checking for absence of KASAN or use-after-free messages; (3) Defender for Cloud or Tenable scan report showing CVE-2026-31657 as resolved — this scan report is the primary remediation closure evidence for audit purposes.

Step 5: Post-Incident — Review whether batman-adv is operationally necessary on affected hosts; disable and blacklist if not in use to reduce kernel attack surface. Evaluate kernel module allowlisting policies. Assess whether Patch Tuesday cadence and Azure Linux update pipelines need tightening to reduce window between patch release and deployment. Map control gap to CIS Benchmark Level 1 for Linux (kernel module restriction controls).

NIST Phase: Post Incident

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

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST CM-7 (Least Functionality), NIST SI-2 (Flaw Remediation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: To enforce batman-adv blacklisting at scale without a configuration management platform, deploy a bash script via cron or cloud-init: `echo 'install batman_adv /bin/false' >> /etc/modprobe.d/disable-batman-adv.conf` — the `install /bin/false` directive prevents both manual and automatic loading more robustly than `blacklist` alone. For kernel module allowlisting without a commercial tool, configure module signing enforcement: set `MODULE_SIG_FORCE=y` in kernel build config, or on existing systems document the inventory of required modules and diff against `lsmod` output weekly via cron. File the results in a shared drive as audit evidence for the lessons-learned report.

Evidence: For the post-incident review record, compile: (1) the full timeline from MSRC April 2026 Patch Tuesday release date to deployment completion across all affected hosts, quantifying the exposure window for CVE-2026-31657; (2) the pre-remediation inventory of hosts where batman-adv was loaded but mesh networking was not a documented operational requirement — this gap represents an unmanaged kernel attack surface; (3) the `/etc/modprobe.d/` configuration state before and after remediation to document the hardening delta applied as a result of this incident.

Detection Guidance

Query asset inventory for Azure Linux 3.0 hosts running kernel version 6.6.130.1-3 (command: 'uname -r' or MDE/Defender for Cloud device inventory filtered by OS version). Check if batman-adv module is loaded: 'lsmod | grep batman_adv'. For fleet-level detection, use Defender for Cloud's CVE recommendations filtered on CVE-2026-31657, or consult Tenable Nessus plugin database for current coverage of CVE-2026-31657. Monitor kernel ring buffer ('dmesg') and /var/log/kern.log for use-after-free indicators: 'BUG: KASAN: use-after-free', 'general protection fault', or unexpected kernel oops entries referencing batman_adv. No public IOC signatures or exploit code have been confirmed at time of writing; behavioral detection should focus on kernel-level anomalies and unauthorized privilege escalation events (monitor for unexpected root process spawning from network-adjacent services).

Framework Mappings

MITRE-ATTACK

- **T1499.004** — Application or System Exploitation
- **T1068** — Exploitation for Privilege Escalation

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-16** — Memory Protection

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
- **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
T1499.004	Application or System Exploitation	Impact
T1068	Exploitation for Privilege Escalation	Privilege-Escalation

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31657	T1
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Apr	T1
CVE-2026-31657 - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-31657	T1
CVE-2026-31657 Common Vulnerabilities and Exposures | SUSE	https://www.suse.com/security/cve/CVE-2026-31657.html	T3
Linux Distros Unpatched Vulnerability : CVE-2026-31657 | Tenable®	https://www.tenable.com/plugins/nessus/310309	T3

Generated 2026-04-29 06:52 UTC by TJS Security Command Center