INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-29 06:53 UTC

CVE-2026-31659: batman-adv Oversized Global TT Response Buffer Rejection, Critical Kernel Vulnerability in Azure Linux 3.0

CVE VULNERABILITY | **CRITICAL** | CVSS 9.8

SCC Item ID	SCC-CVE-2026-0090
Type	CVE Vulnerability
CVE ID	CVE-2026-31659
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.0005 (17th percentile)
Affected Products	Microsoft Azure Linux 3.0, azl3 kernel 6.6.130.1-3
Published	2026-04-29T01:47:49
Discovery Source	Msrc Patch Tuesday

Executive Summary

A critical vulnerability (CVSS 9.8) in the batman-adv kernel module affects Microsoft Azure Linux 3.0 systems running kernel version 6.6.130.1-3 or earlier. The flaw is remotely exploitable with no authentication required, allowing an attacker to trigger memory corruption via malformed mesh networking packets. Microsoft addressed this in the April 2026 Patch Tuesday release; organizations running Azure Linux 3.0 workloads should treat this as a priority patching event.

Technical Analysis

CVE-2026-31659 is a buffer overflow and improper buffer size calculation vulnerability (CWE-120, CWE-131) in the batman-adv kernel module, which implements the B.A.T.M.A.N. Layer 2 mesh networking protocol included in the mainline Linux kernel. The flaw resides in the Global Translation Table (TT) response handling path: the module fails to reject oversized TT response buffers before processing, creating a memory safety violation. CVSS 9.8 with a network attack vector (AV:N), low complexity (AC:L), and no privileges or user interaction required indicates unauthenticated remote exploitability. Affected scope is Microsoft Azure Linux 3.0, specifically the azl3 kernel package at version 6.6.130.1-3 and prior. MITRE technique mappings include T1210 (Exploitation of Remote Services) and T1499.004 (Application or System Exploitation). The upstream fix enforces buffer size validation before the TT response is processed. EPSS score is 0.00053 (16.5th percentile), indicating low observed exploitation probability at time of disclosure; the vulnerability is not currently listed on

the CISA KEV catalog. Patch delivered via Microsoft MSRC April 2026 update cycle. Sources: MSRC (T1), NVD (T1).

Action Checklist

1. Step 1: Containment, Identify all Azure Linux 3.0 hosts running kernel versions prior to the patched azl3 package. If batman-adv is not required for operations, unload the module immediately: 'rmmod batman-adv' and add it to the kernel module blacklist (/etc/modprobe.d/blacklist.conf) to prevent reload on reboot. If the module cannot be disabled, isolate affected hosts from untrusted network segments at the network layer.
2. Step 2: Detection, Query your asset inventory or CMDB for Azure Linux 3.0 hosts. On each host, confirm kernel version via 'uname -r' and check whether batman-adv is loaded via 'lsmod | grep batman_adv'. Review network traffic logs for anomalous Layer 2 mesh protocol activity or unexpected batman-adv (EtherType 0x4305) frames originating from external or untrusted sources. Check kernel logs (/var/log/kern.log or 'dmesg') for memory fault or OOM events near batman-adv module entries.
3. Step 3: Eradication, Apply the Microsoft April 2026 Patch Tuesday update for Azure Linux 3.0. The target package is the azl3 kernel at the patched version released under CVE-2026-31659. Update via the standard Azure Linux package manager ('tdnf update kernel'). Verify the installed kernel version post-update. Reference: MSRC Update Guide at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31659 (T1 source, human validation recommended before use).
4. Step 4: Recovery, After patching and reboot, confirm the running kernel version reflects the patched release. Re-enable batman-adv only if operationally required, and verify module load against the patched kernel. Monitor kernel logs and network telemetry for 24-48 hours post-patch for anomalous Layer 2 traffic or memory fault indicators. Validate that affected hosts return clean results in your vulnerability scanner.
5. Step 5: Post-Incident, Assess whether batman-adv is required across your Azure Linux 3.0 fleet; if not, add permanent module blacklisting to your hardening baseline. Review your patch SLA for CVSS 9.x kernel vulnerabilities; this item should have triggered an expedited patch cycle. If patch deployment took longer than 72 hours from advisory publication, identify the process gap. Map this finding to CIS Benchmark controls for kernel module management and update your Azure Linux hardening standard accordingly.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and legal/compliance if: (1) any Azure Linux 3.0 host running kernel 6.6.130.1-3 or earlier is confirmed to process PII, PHI, or PCI-DSS cardholder data and was reachable from an untrusted network segment prior to containment, triggering breach notification assessment; (2) kernel BUG, KASAN, or general protection fault entries in dmesg are temporally correlated with inbound EtherType 0x4305 frames from external sources, indicating active exploitation rather than theoretical exposure; or (3) the team lacks the capability to enumerate all Azure Linux 3.0 hosts fleet-wide within 4 hours, indicating an asset inventory gap requiring executive resource authorization.

Recovery Notes	Post-patch recovery validation must confirm the running kernel via 'uname -r' matches the patched azl3 release published under CVE-2026-31659, and that either batman-adv is absent from 'lsmod' output or is loading against the patched kernel binary — not the vulnerable 6.6.130.1-3 version. Monitor kernel logs (/var/log/kern.log and journalctl -k) and network telemetry for EtherType 0x4305 frames for a minimum of 48 hours post-patch, as memory corruption exploitation of this class may produce delayed crash artifacts or deferred OOM events. Any host that exhibited pre-patch kernel BUG or KASAN entries correlated with batman-adv activity should be treated as potentially compromised and considered for full OS rebuild from a known-good Azure Linux 3.0 image before returning to production.
Forensic Artifacts	dmesg and /var/log/kern.log entries containing 'batman_adv' co-occurring with 'BUG:', 'KASAN:', 'general protection fault', 'Unable to handle kernel NULL pointer dereference', or 'kernel stack overflow' — these would be the direct kernel-level signature of the CVE-2026-31659 memory corruption trigger via an oversized Global TT Response packet | Network packet capture (pcap) filtered for EtherType 0x4305 frames on all interfaces — specifically frames originating from MAC addresses not belonging to known internal mesh peers, or frames with abnormally large TT response payloads that exceed expected batman-adv protocol limits, indicating crafted exploitation traffic | /sys/kernel/debug/batman_adv/bat0/transtable_global — the in-memory global translation table maintained by batman-adv; an attacker exploiting the oversized TT response buffer flaw would manipulate this table, and its state at time of incident provides direct evidence of attempted or successful exploitation | tdnf history output ('tdnf history' or /var/log/tdnf.log) showing the kernel package version installed on each Azure Linux 3.0 host, establishing which hosts ran the vulnerable azl3 kernel 6.6.130.1-3 and for how long prior to patching — critical for breach scope determination | VPC flow logs or cloud network logs filtered for Layer 2 / mesh protocol activity targeting Azure Linux 3.0 host IP addresses from external or cross-segment sources during the window between MSRC advisory publication (April 2026 Patch Tuesday) and confirmed containment — absence of such traffic reduces exploitation likelihood; presence requires escalation and host forensic triage

Per-Action IR Details

Step 1: Containment — Identify all Azure Linux 3.0 hosts running kernel versions prior to the patched azl3 package. If batman-adv is not required for operations, unload the module immediately: 'rmmod batman-adv' and add it to the kernel module blacklist (/etc/modprobe.d/blacklist.conf) to prevent reload on reboot. If the module cannot be disabled, isolate affected hosts from untrusted network segments at the network layer.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST CM-7 (Least Functionality), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.8 (Uninstall or Disable Unnecessary Services on Enterprise Assets and Software)

Compensating: For teams without centralized orchestration: deploy a two-liner bash script via SSH to enumerate and unload batman-adv across the fleet — 'for host in $(cat azure_linux_hosts.txt); do ssh $host "lsmod | grep batman_adv && rmmod batman-adv && echo 'batman-adv' >> /etc/modprobe.d/blacklist.conf"; done'. For network-layer isolation where module cannot be disabled, use iptables to drop inbound EtherType 0x4305 frames: 'iptables -I INPUT -p 0x4305 -j DROP' (requires kernel support for raw EtherType matching via arptables or ebtables on the bridge interface). Verify blacklist persistence with 'cat /etc/modprobe.d/blacklist.conf | grep batman'.

Evidence: Before unloading the module, capture a memory snapshot using 'avml' or '/proc/kcore' (if accessible) to preserve any in-memory evidence of prior exploitation attempts. Record current module state with 'lsmod > /tmp/lsmod_pre_remediation_$(hostname)_$(date +%Y%m%d%H%M%S).txt'. Capture batman-adv runtime state from /sys/kernel/debug/batman_adv/ (if debugfs is mounted) — specifically the 'tt_global_entries' and 'tt_local_entries' tables, which would reflect oversized TT response injection attempts. Dump dmesg output immediately: 'dmesg -T >

/tmp/dmesg_pre_remediation_$(hostname).txt' and preserve any entries referencing 'batman_adv', 'BUG:', 'kernel NULL pointer dereference', 'general protection fault', or 'KASAN' (Kernel Address Sanitizer) stack traces that would indicate triggered memory corruption.

Step 2: Detection — Query your asset inventory or CMDB for Azure Linux 3.0 hosts. On each host, confirm kernel version via 'uname -r' and check whether batman-adv is loaded via 'lsmod | grep batman_adv'. Review network traffic logs for anomalous Layer 2 mesh protocol activity or unexpected batman-adv (EtherType 0x4305) frames originating from external or untrusted sources. Check kernel logs (/var/log/kern.log or 'dmesg') for memory fault or OOM events near batman-adv module entries.

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 AU-2 (Event Logging), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 8.2 (Collect Audit Logs)

Compensating: Use osquery to enumerate kernel version and loaded modules fleet-wide without a CMDB: 'osqueryi "SELECT version FROM kernel_info; SELECT name, path FROM kernel_modules WHERE name LIKE '%batman%';"'. For network-layer detection without a SIEM, use Wireshark or tcpdump on a network tap or mirror port to filter for batman-adv frames: 'tcpdump -i eth0 -nn ether proto 0x4305 -w /tmp/batmanadv_capture_$(date +%Y%m%d).pcap'. Write a Sigma rule targeting syslog/kern.log for the strings 'batman_adv' co-occurring with 'general protection fault', 'BUG:', 'KASAN', or 'Unable to handle kernel' within a 60-second window. Parse /var/log/kern.log with: 'grep -E "batman_adv|BUG:|KASAN|general protection fault|kernel NULL pointer" /var/log/kern.log | grep -v "^#"'.

Evidence: Collect pcap files from network taps or cloud VPC flow logs filtered for EtherType 0x4305 traffic, noting source MAC addresses and originating network segments — external or cross-VLAN batman-adv frames are anomalous in Azure Linux environments not intentionally running mesh networks. Export full dmesg output timestamped around any detected batman-adv activity. From /proc/net/, capture 'softnet_stat' and 'dev' to identify which interface received anomalous Layer 2 frames. Retrieve /sys/kernel/debug/batman_adv/bat0/transtable_global if the interface is active — an oversized TT global response table would be a direct indicator of attempted exploitation of the buffer rejection flaw. Collect 'journalctl -k --since="48 hours ago" > /tmp/kernel_journal_$(hostname).txt' to capture structured kernel log entries that survive rotation.

Step 3: Eradication — Apply the Microsoft April 2026 Patch Tuesday update for Azure Linux 3.0. The target package is the azl3 kernel at the patched version released under CVE-2026-31659. Update via the standard Azure Linux package manager ('tdnf update kernel' or equivalent). Verify the installed kernel version post-update. Reference: MSRC Update Guide at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31659 (T1 source, human validation recommended before use).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-3 (Configuration Change Control), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: For teams without an enterprise patch management platform, use a coordinated bash script executed via Ansible or direct SSH to apply the tdnf update and capture pre/post kernel versions: 'ssh $host "uname -r > /tmp/kernel_before.txt && tdnf update kernel -y && uname -r > /tmp/kernel_after.txt"'. Validate package integrity after download using the RPM GPG signature check built into tdnf: 'tdnf check-update kernel && rpm -K /var/cache/tdnf/packages/kernel-*.rpm'. For hosts where a reboot is deferred, confirm the patched package is staged and the old kernel is no longer the default boot entry: 'grubby --default-kernel' should reflect the patched version. Document the delta between 'kernel_before.txt' and 'kernel_after.txt' for each host as your patch verification record.

Evidence: Before applying the patch, capture a full package manifest: 'rpm -qa > /tmp/rpm_manifest_pre_patch_$(hostname)_$(date +%Y%m%d).txt'. Record the exact vulnerable kernel version string from 'uname -r' and 'rpm -q kernel'. If any prior exploitation is suspected based on Step 2 findings (kernel BUG/KASAN entries, anomalous EtherType 0x4305 traffic), capture a full disk image or at minimum a targeted forensic acquisition of

/var/log/, /proc/net/, and /sys/kernel/debug/batman_adv/ before patching, as the patch process and reboot will overwrite in-memory evidence and may rotate logs. Preserve the vulnerable kernel binary at /boot/vmlinuz-6.6.130.1-3 (or equivalent) for potential later analysis against known-good hashes from the Azure Linux package repository.

Step 4: Recovery — After patching and reboot, confirm the running kernel version reflects the patched release. Re-enable batman-adv only if operationally required, and verify module load against the patched kernel. Monitor kernel logs and network telemetry for 24-48 hours post-patch for anomalous Layer 2 traffic or memory fault indicators. Validate that affected hosts return clean results in your vulnerability scanner.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST SI-4 (System Monitoring), NIST SI-6 (Security and Privacy Function Verification), NIST IR-5 (Incident Monitoring), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 4.2 (Establish and Maintain a Secure Configuration Process for Network Infrastructure)

Compensating: Post-reboot validation script for a 2-person team: 'uname -r' must reflect the patched kernel version published under CVE-2026-31659; 'lsmod | grep batman_adv' must return empty if blacklisted. For continuous 24-48 hour monitoring without a SIEM, set up a cron job to check kernel logs every 5 minutes and alert on batman-adv or memory fault strings: 'echo "*/5 * * * * root grep -qE \"batman_adv|BUG:|KASAN\" /var/log/kern.log && logger -t CVE-2026-31659-MONITOR ALERT" >> /etc/cron.d/cve_monitor'. Run OpenVAS or a Greenbone Community Edition scan targeting the patched hosts using the CVE-2026-31659 plugin once available, or manually confirm kernel version via authenticated scan. Re-run the tcpdump EtherType 0x4305 capture from Step 2 for 30 minutes post-recovery to verify no residual mesh protocol traffic.

Evidence: Capture a post-patch package manifest ('rpm -qa > /tmp/rpm_manifest_post_patch_$(hostname).txt') and diff against the pre-patch manifest from Step 3 to confirm only the expected kernel packages changed. Document 'uname -r' output with timestamp as the primary patch verification artifact. If batman-adv is re-enabled, capture the module load event from journald: 'journalctl -k | grep batman_adv' and verify the module is loading against the patched kernel version, not the vulnerable 6.6.130.1-3 binary. Retain all kernel logs collected during the 24-48 hour monitoring window as evidence that no post-patch exploitation indicators emerged.

Step 5: Post-Incident — Assess whether batman-adv is required across your Azure Linux 3.0 fleet; if not, add permanent module blacklisting to your hardening baseline. Review your patch SLA for CVSS 9.x kernel vulnerabilities; this item should have triggered an expedited patch cycle. If patch deployment took longer than 72 hours from advisory publication, identify the process gap. Map this finding to CIS Benchmark controls for kernel module management and update your Azure Linux hardening standard accordingly.

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 7.2 (Establish and Maintain a Remediation Process), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: For teams without a formal GRC platform, document the lessons learned in a structured markdown post-mortem covering: (1) time from MSRC advisory publication to fleet-wide patch completion, (2) which hosts required manual intervention vs. automated tdnf update, (3) whether batman-adv was confirmed operationally necessary on any host. Encode the batman-adv blacklist as a hardening script committed to version control: 'echo "blacklist batman_adv" >> /etc/modprobe.d/cve-2026-31659-hardening.conf && depmod -a'. Submit the blacklist configuration as a pull request to your Azure Linux golden image pipeline. Create a YARA rule or Sigma detection targeting future CVEs in the batman-adv module family for proactive threat hunting.

Evidence: Compile a full timeline artifact: MSRC advisory publication date (April 2026 Patch Tuesday) versus first detection in your environment, versus containment timestamp (rmmod execution), versus patch completion timestamp across all hosts. Retrieve patch deployment logs from tdnf history: 'tdnf history' on each host or from your configuration management tooling. Preserve the diff between pre- and post-patch hardening baselines as documentary evidence for audit purposes. If this CVE affected hosts processing regulated data (PII, PHI, PCI-DSS cardholder data), retain all IR artifacts per your data breach notification retention policy — CVSS 9.8 with remote unauthenticated exploitation on

production Azure Linux workloads may trigger breach assessment obligations depending on workload classification.

Detection Guidance

Check for batman-adv module presence: run 'lsmod | grep batman_adv' on Azure Linux 3.0 hosts. Confirm kernel version with 'uname -r', any version prior to the April 2026 patched azl3 release is vulnerable. In network monitoring tools, filter for EtherType 0x4305 (batman-adv OGM packets) or unusual Layer 2 broadcast traffic not consistent with your topology. In kernel logs ('dmesg' or /var/log/kern.log), look for entries referencing 'batman-adv', 'tt_global', buffer allocation failures, or kernel OOP (Oops) traces near batman-adv stack frames. If SIEM is ingesting syslog from Linux hosts, query for source 'kernel' with message fields containing 'batman' or 'tt_global_response'. No public IOCs or exploitation indicators are available at this time; detection is exposure-focused rather than artifact-focused.

Framework Mappings

MITRE-ATTACK

- **T1499.004** — Application or System Exploitation
- **T1210** — Exploitation of Remote Services

NIST-800-53R5

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

OWASP-TOP10-2021

- **A03:2021** — Injection

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1499.004	Application or System Exploitation	Impact

Technique ID	Technique Name	Tactic
T1210	Exploitation of Remote Services	Lateral-Movement

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31659	T1
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Apr	T1
CVE-2026-31659 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-31659	T1
CVE-2026-31659 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-31659	T3
CVE-2026-31659 - Vulnerability Details - OpenCVE	https://app.opencve.io/cve/CVE-2026-31659	T3

Generated 2026-04-29 06:53 UTC by TJS Security Command Center