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

CVE-2026-31669: Critical MPTCP Slab-Use-After-Free in Linux Kernel Affects Azure Linux 3.0

CVE VULNERABILITY | CRITICAL | CVSS 9.8

SCC Item ID	SCC-CVE-2026-0089
Type	CVE Vulnerability
CVE ID	CVE-2026-31669
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.0007 (21th 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 memory corruption flaw (CVE-2026-31669, CVSS 9.8) in the Linux kernel's Multipath TCP subsystem affects Microsoft Azure Linux 3.0 systems running kernel version 6.6.130.1-3, disclosed in Microsoft's April 2026 Patch Tuesday. The vulnerability is remotely exploitable with no credentials or user interaction required, meaning exposed Azure Linux 3.0 workloads could be compromised without any insider access. Organizations running unpatched Azure Linux 3.0 infrastructure face risk of arbitrary code execution in kernel context with root privileges, which can cascade to data exfiltration, service outages, and lateral movement across cloud environments.

Technical Analysis

CVE-2026-31669 is a slab-use-after-free vulnerability (CWE-416) in the Linux kernel MPTCP subsystem, specifically within the __inet_lookup_established() function. Use-after-free flaws in kernel networking paths occur when a freed memory region is dereferenced before reallocation, allowing an attacker to corrupt kernel heap structures or redirect execution flow. CVSS base score is 9.8 (Critical) with an attack vector of Network, low complexity, no privileges required, and no user interaction, consistent with remote kernel exploitation potential. Affected platform: Microsoft Azure Linux 3.0, azl3 kernel package version 6.6.130.1-3. Red Hat and SUSE have issued advisories for the same upstream CVE, per vendor security releases, indicating the flaw

exists in the broader Linux 6.6.x kernel tree. EPSS score is 0.068% (20.6th percentile) as of disclosure, indicating limited observed exploitation activity at this time. MITRE ATT&CK techniques: T1068 (Exploitation for Privilege Escalation) and T1499.004 (Application or System Exploitation). The CVE is not currently listed on CISA's Known Exploited Vulnerabilities catalog. Patch source: Microsoft Security Response Center April 2026 update guide (azl3 kernel package update). Red Hat and SUSE advisories confirm upstream kernel 6.6.x exposure.

Action Checklist

1. Step 1: Containment. Identify all Azure Linux 3.0 systems running azl3 kernel 6.6.130.1-3 in your environment. Prioritize internet-facing or multi-tenant workloads. Where patching is not immediately possible, consider restricting inbound MPTCP traffic at the network perimeter or disabling MPTCP if operationally feasible. Verify whether any systems also run Red Hat or SUSE Linux on kernel 6.6.x and apply the same exposure triage.
2. Step 2: Detection. Query your asset inventory and cloud management plane for Azure Linux 3.0 instances. Use 'uname -r' or equivalent CMDB queries to confirm kernel version 6.6.130.1-3. Review kernel and system logs (/var/log/kern.log, /var/log/syslog) for memory corruption or allocator error messages referencing mptcp or inet_lookup_established, which may indicate exploitation attempts. Check Azure Monitor and Defender for Cloud for anomalous kernel-level activity on affected hosts.
3. Step 3: Eradication. For Azure Linux 3.0: apply the updated azl3 kernel package as released in the Microsoft April 2026 Patch Tuesday update using 'tdnf update kernel'. For Red Hat systems on kernel 6.6.x: use 'dnf update kernel' with the published errata. For SUSE systems: use 'zypper update kernel' with the published errata. Confirm the updated kernel is loaded post-reboot with 'uname -r'.
4. Step 4: Recovery. After patching and rebooting, confirm the running kernel version no longer matches 6.6.130.1-3. Re-enable MPTCP if it was disabled as a temporary control, and verify application functionality. Monitor affected systems for 48-72 hours post-patch using kernel integrity tools (e.g., auditd, Falco) for anomalous process spawning, privilege escalation attempts, or unexpected network socket behavior that could indicate pre-patch exploitation.
5. Step 5: Post-Incident. Evaluate whether your patch cadence for Azure Linux 3.0 workloads meets the risk profile of Critical-rated kernel CVEs. Assess whether MPTCP is required in your environment; if not, disabling it reduces kernel attack surface permanently. Review cloud workload protection (CWP) coverage for Azure Linux 3.0 hosts. Confirm that kernel-level CVEs from upstream Linux 6.6.x are tracked across all distributions in your fleet, not only the Microsoft-branded advisory.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and cloud security leadership immediately if any host running azl3 kernel 6.6.130.1-3 shows KASAN slab-use-after-free messages referencing mptcp or inet_lookup_established in kern.log, unexpected privilege escalation events in auditd or Falco output post-exploitation window, or if the affected workload processes PII, PHI, or financial data triggering breach notification obligations under GDPR, HIPAA, or PCI DSS.

Recovery Notes	After patching all Azure Linux 3.0 hosts to the post-6.6.130.1-3 azl3 kernel from the Microsoft April 2026 Patch Tuesday update, monitor kernel logs (/var/log/kern.log) and Falco/auditd output continuously for 48-72 hours for any residual KASAN memory corruption messages, unexpected MPTCP subflow establishments, or privilege escalation events that would indicate a pre-patch compromise persisted through the kernel upgrade. If any host was identified as potentially exploited prior to patching (KASAN errors present, anomalous network sockets, or unauthorized processes), treat it as a confirmed incident: do not return it to production without a full memory forensic analysis using LiME and verification that no kernel rootkit or rogue module is loaded ('lsmod | diff - /tmp/lsmod-baseline.txt'). Re-enable MPTCP only on hosts where it is operationally required and only after the patched kernel is confirmed running.
Forensic Artifacts	/var/log/kern.log and dmesg output: KASAN slab-use-after-free and SLUB allocator error messages containing 'mptcp', 'inet_lookup_established', or 'use-after-free' are the primary kernel-emitted indicators that CVE-2026-31669 was triggered on a specific host. | /var/crash/ kernel crash dumps (kdump): If the MPTCP slab-use-after-free caused a kernel panic, kdump will have captured a vmcore file preserving the full kernel memory state at the time of crash, including the corrupted slab object and call stack pointing into the MPTCP subsystem. | LiME (Linux Memory Extractor) full RAM capture: A live memory image taken from a potentially exploited host will preserve any attacker-injected kernel shellcode, rogue kernel module, or modified MPTCP socket structures in the slab allocator that would not survive a reboot or be visible in filesystem forensics. | /proc/net/mptcp and 'ss -tnp' output: Active MPTCP subflow records at time of incident capture document any remote IP addresses that established MPTCP connections to the vulnerable host, providing network IOCs for threat actor infrastructure correlation via MITRE ATT&CK T1071 (Application Layer Protocol) and T1595 (Active Scanning). | Azure Defender for Cloud and Azure Monitor alert telemetry: Cloud-native detections for the affected VM resource ID, specifically alerts categorized under anomalous kernel activity or Linux privilege escalation, provide a cloud-plane corroborating record independent of host-based log tampering if an attacker achieved root post-exploitation.

Per-Action IR Details

Step 1: Containment — Identify all Azure Linux 3.0 systems running azl3 kernel 6.6.130.1-3 in your environment. Prioritize internet-facing or multi-tenant workloads. Where patching is not immediately possible, consider restricting inbound MPTCP traffic at the network perimeter or disabling MPTCP if operationally feasible (sysctl net.mptcp.enabled=0). Verify whether any systems also run Red Hat or SUSE Linux on kernel 6.6.x and apply the same exposure triage.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Run 'uname -r' across all Azure Linux 3.0 hosts via a parallel SSH loop: 'for h in $(cat hosts.txt); do ssh $h uname -r; done \| grep 6.6.130.1-3' to enumerate exposed nodes. Use Azure CLI to enumerate VMs: 'az vm list --query "[].{name:name, os:storageProfile.imageReference.offer}" -o table'. Block inbound MPTCP (TCP option kind 30) at the perimeter firewall with an iptables rule: 'iptables -A INPUT -p tcp --tcp-option 30 -j DROP'. Apply 'sysctl -w net.mptcp.enabled=0' immediately on all confirmed vulnerable hosts and persist via /etc/sysctl.d/99-disable-mptcp.conf.

Evidence: Before containing, capture the current MPTCP socket state and active connection table from affected hosts: run 'ss -tnp \| grep -i mptcp' and 'cat /proc/net/mptcp' to document any active MPTCP sessions that may represent an in-progress exploitation channel. Capture kernel ring buffer output via 'dmesg \| grep -iE "mptcp\|use-after-free\|KASAN\|slab\|inet_lookup"' and save to a timestamped file as evidence of any prior memory

corruption events before the host is modified.

Step 2: Detection — Query your asset inventory and cloud management plane for Azure Linux 3.0 instances. Use 'uname -r' or equivalent CMDB queries to confirm kernel version 6.6.130.1-3. Review kernel and system logs (/var/log/kern.log, /var/log/syslog) for KASAN or SLUB allocator error messages referencing mptcp or inet_lookup_established, which may indicate exploitation attempts or memory corruption events. Check Azure Monitor and Defender for Cloud for anomalous kernel-level activity on affected hosts.

NIST Phase: Detection Analysis

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

Controls: NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring), NIST AU-2 (Event Logging), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Use grep to scan kernel logs for KASAN slab-use-after-free indicators specific to MPTCP: 'grep -iE "KASAN|use-after-free|mptcp|inet_lookup_established|BUG: unable to handle" /var/log/kern.log /var/log/syslog > /tmp/cve-2026-31669-indicators.txt'. Deploy the Falco rule targeting unexpected kernel module loads or privilege escalation post-MPTCP socket handling: write a custom Falco rule watching for processes gaining CAP_SYS_ADMIN or CAP_NET_ADMIN spawned within 30 seconds of a new MPTCP subflow establishment. Use 'ausearch -m SYSCALL -sv no' with auditd to surface failed syscalls that may indicate exploit instability or crash-loop behavior.

Evidence: Collect /var/log/kern.log and /var/log/syslog entries containing 'KASAN', 'slab-out-of-bounds', 'use-after-free', 'mptcp', or 'inet_lookup_established' — these strings are directly emitted by the Linux kernel's KASAN memory error detector when the slab-use-after-free in the MPTCP subsystem is triggered. Also collect 'dmesg' output in full, kernel crash dumps from /var/crash/ (if kdump is enabled), and Azure Defender for Cloud alerts under the 'Unusual kernel activity' or 'Potential exploitation of Linux kernel vulnerability' categories for the affected host's resource ID.

Step 3: Eradication — Apply the updated azl3 kernel package as released in the Microsoft April 2026 Patch Tuesday update. Use 'tdnf update kernel' on Azure Linux 3.0 hosts or deploy via your standard patch management tooling. For Red Hat and SUSE systems on kernel 6.6.x, apply the respective vendor kernel errata as published in their CVE-2026-31669 advisories. Confirm the updated kernel is loaded post-reboot with 'uname -r'.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-3 (Configuration Change Control), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: On each Azure Linux 3.0 host, run 'tdnf update kernel -y && reboot' and capture pre/post kernel version via 'uname -r > /tmp/kernel-pre-patch.txt' before patching. Verify package integrity post-install with 'rpm -V kernel' to confirm the installed files match the signed package manifest from the Microsoft azl3 repository. For Red Hat 6.6.x hosts, apply the CVE-2026-31669 errata using 'yum update kernel --advisory '; for SUSE, use 'zypper patch --cve CVE-2026-31669'. Document each host's pre- and post-patch kernel version in a remediation tracking spreadsheet.

Evidence: Before rebooting into the patched kernel, image the running system's volatile state if exploitation is suspected: capture full memory with LiME (Linux Memory Extractor, a free kernel module) via 'insmod lime.ko path=/mnt/evidence/mem.lime format=lime' to preserve any attacker-resident code or MPTCP subflow state in the slab allocator. Capture the installed package list ('rpm -qa | sort > /tmp/packages-pre-patch.txt') and running process tree ('ps auxf > /tmp/proctree-pre-patch.txt') to establish a pre-patch baseline for later comparison if post-patch anomalies emerge.

Step 4: Recovery — After patching and rebooting, confirm the running kernel version no longer matches 6.6.130.1-3. Re-enable MPTCP if it was disabled as a temporary control, and verify application functionality. Monitor affected systems for 48-72 hours post-patch using kernel integrity tools (e.g., auditd, Falco) for

anomalous process spawning, privilege escalation attempts, or unexpected network socket behavior that could indicate pre-patch exploitation.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-6 (Security and Privacy Function Verification), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST CP-10 (System Recovery and Reconstitution), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Confirm patched kernel with 'uname -r' and verify MPTCP subsystem loads cleanly via 'dmesg | grep -i mptcp' — absence of KASAN errors on subflow establishment confirms the patched code path is active. Re-enable MPTCP with 'sysctl -w net.mptcp.enabled=1' only after confirming 'uname -r' shows a version beyond 6.6.130.1-3. Deploy Falco with the syscall ruleset to monitor for privilege escalation (setuid/setgid calls, CAP_SYS_ADMIN grants) and unexpected outbound connections from kernel threads for the 48-72 hour window. Use 'auditctl -a always,exit -F arch=b64 -S execve -k post_patch_exec' to log all process executions for post-patch baselining.

Evidence: Post-reboot, run 'rpm -V kernel' again on the patched system and diff against the pre-patch package verification output to confirm no files were left in a modified state by a potential pre-patch compromise. Collect a post-patch memory snapshot with LiME and compare running process list ('ps auxf') against the pre-patch baseline to detect any persistent attacker implants (kernel rootkits or rogue kernel modules) that would survive the kernel upgrade if the system was compromised prior to patching.

Step 5: Post-Incident — Evaluate whether your patch cadence for Azure Linux 3.0 workloads meets the risk profile of Critical-rated kernel CVEs. Assess whether MPTCP is required in your environment; if not, disabling it reduces kernel attack surface permanently. Review cloud workload protection (CWP) coverage for Azure Linux 3.0 hosts. Confirm that kernel-level CVEs from upstream Linux 6.6.x are tracked across all distributions in your fleet — not only the Microsoft-branded advisory.

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 RA-3 (Risk Assessment), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST CM-6 (Configuration Settings), 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: Establish a standing osquery scheduled query to detect kernel version drift across your Azure Linux 3.0 fleet: 'SELECT name, version FROM kernel_info WHERE version LIKE "6.6.%"' run daily via osquery's schedule config. If MPTCP is confirmed unnecessary, persist its disablement by writing 'net.mptcp.enabled=0' to /etc/sysctl.d/99-disable-mptcp.conf and verify it survives reboot. Subscribe directly to the upstream Linux 6.6.x stable kernel mailing list (kernel.org) and MSRC (Microsoft Security Response Center) Azure Linux advisories so cross-distribution kernel CVEs — such as those affecting both azl3 and RHEL/SUSE 6.6.x trees — are captured before vendor-specific advisories are published.

Evidence: Document lessons-learned artifacts specific to CVE-2026-31669: the full timeline from Microsoft April 2026 Patch Tuesday disclosure to patch deployment on each host (as a patch SLA measurement), any KASAN or SLUB log entries collected during the detection phase (as indicators of exposure or exploitation), and the pre/post kernel version records from each host. These artifacts satisfy NIST IR-5 (Incident Monitoring) documentation requirements and provide the evidentiary basis for any post-incident risk assessment update or cloud workload protection gap analysis.

Detection Guidance

No public exploit code or active exploitation indicators (IOCs) are confirmed for CVE-2026-31669 at this time. Detection should focus on exposure confirmation and behavioral anomaly monitoring. (1) Asset identification: run 'uname -r' across Azure Linux 3.0 hosts; flag any returning 6.6.130.1-3. (2) Kernel error telemetry: search syslog and kern.log for strings containing 'use-after-free' or 'mptcp', which may indicate exploitation attempts.

Note: KASAN and SLUB allocator logs appear only if the kernel is compiled with debugging enabled; production systems may not expose these messages. (3) Behavioral indicators: monitor for unexpected privilege escalation (MITRE T1068), unusual kernel module loads, or new root-level processes spawned from network-facing services. (4) Network telemetry: if MPTCP is in use, log and baseline MPTCP connection establishment patterns; anomalous connection floods or malformed MPTCP options could indicate probing. (5) Cloud telemetry: enable kernel-level anomaly detection rules on Azure Linux 3.0 workloads in Defender for Cloud and Azure Monitor, and review alerts for the patch window period.

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

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
T1068	Exploitation for Privilege Escalation	Privilege-Escalation

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31669	T1
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Apr	T1

Source	URL	Tier
CVE-2026-31669 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-31669	**T1**
CVE-2026-31669 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-31669	**T3**
CVE-2026-31669 Common Vulnerabilities and Exposures | SUSE	https://www.suse.com/security/cve/CVE-2026-31669/	**T3**

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