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

CVE-2026-31478: Critical ksmbd SMB2 Buffer Length Calculation Flaw in Azure Linux 3.0 Kernel

CVE VULNERABILITY | **CRITICAL** | CVSS 9.8

SCC Item ID	SCC-CVE-2026-0091
Type	CVE Vulnerability
CVE ID	CVE-2026-31478
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.0009 (25th 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 (CVE-2026-31478, CVSS 9.8) in the ksmbd SMB2 server component affects the Microsoft Azure Linux 3.0 kernel package (azl3 6.6.130.1-3), disclosed in the April 2026 Microsoft Patch Tuesday release. The flaw allows network-reachable exploitation without authentication, enabling memory corruption on affected Azure Linux 3.0 hosts. Organizations running Azure Linux 3.0 with ksmbd active should treat this as an urgent patching priority.

Technical Analysis

CVE-2026-31478 is a buffer length miscalculation in the ksmbd in-kernel SMB3 server, specifically in the smb2_calc_max_out_buf_len() function. The function uses a hardcoded constant (hdr2_len) instead of the architecturally correct offsetof() macro to compute output buffer boundaries, resulting in incorrect size calculation across hardware platforms. This maps to CWE-119 (Improper Restriction of Operations within the Bounds of a Memory Buffer) and CWE-131 (Incorrect Calculation of Buffer Size). Incorrect boundary values can enable out-of-bounds memory read or write conditions. ksmbd is network-facing by design, making this exploitable without local access, consistent with the 9.8 CVSS base score and MITRE ATT&CK T1210 (Exploitation of Remote Services). Affected package: Microsoft azl3 kernel 6.6.130.1-3 on Azure Linux 3.0. Broader upstream Linux kernel impact is not confirmed in available source data. The fix replaces the hardcoded constant with offsetof(), ensuring accurate struct size computation. EPSS score is 0.00089 (0.25th percentile) as

of disclosure; no active exploitation confirmed and not on CISA KEV. Sources: MSRC Update Guide (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31478), NVD (https://nvd.nist.gov/vuln/detail/CVE-2026-31478).

Action Checklist

1. **Containment:** Identify all Azure Linux 3.0 hosts running the azl3 kernel package at version 6.6.130.1-3. If ksmbd is active and the service is network-reachable, restrict SMB (TCP 445) access at the network perimeter or host firewall to trusted sources only until the patch is applied.
2. **Detection:** Query your asset inventory and CMDB for Azure Linux 3.0 instances. On candidate hosts, run 'uname -r' to confirm kernel version and 'lsmod | grep ksmbd' or 'systemctl status ksmbd' to determine if ksmbd is loaded and running. Review network flow logs for unexpected SMB traffic (TCP/445) inbound to Azure Linux hosts.
3. **Eradication:** Apply the patched azl3 kernel package via the standard Azure Linux package manager (tdnf update kernel) or through your Azure Linux patching pipeline. Confirm the updated package version supersedes 6.6.130.1-3. Reference the MSRC April 2026 update guide for the specific fixed build identifier.
4. **Recovery:** After patching, reboot affected hosts to load the updated kernel. Re-verify kernel version with 'uname -r'. Confirm ksmbd behavior is normal if the service is required. Re-enable any SMB firewall rules that were restricted during containment. Monitor SMB-facing logs for anomalous connection attempts for 72 hours post-patch.
5. **Post-Incident:** Assess whether ksmbd is operationally required on all affected hosts; disable the service where it is not needed to reduce attack surface. Review kernel patch SLA adherence for CVSS Critical findings. Evaluate whether Azure Linux 3.0 hosts with network-facing kernel services are covered by your vulnerability scanning and detection rule inventory.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and activate full IR procedures immediately if: kernel OOPS, BUG, or NULL pointer dereference entries referencing ksmbd are found in '/var/log/kern.log' or '/var/crash/' on any Azure Linux 3.0 host (indicating the vulnerability was triggered), any unexpected authenticated or unauthenticated SMB session is established from an external or non-standard internal IP to a host running azl3 6.6.130.1-3, or if PII/PHI data is stored on file shares hosted by the affected ksmbd service (triggering breach notification assessment under applicable regulatory frameworks).
Recovery Notes	After rebooting to the patched azl3 kernel, confirm via 'uname -r' that the loaded kernel version matches the MSRC April 2026 fixed build identifier before re-enabling any SMB firewall rules. Monitor '/var/log/kern.log' and inbound TCP 445 network captures for 72 hours post-patch, paying specific attention to repeat connection attempts from any source IPs observed in pre-patch SMB flow logs — these would indicate a threat actor with prior knowledge of the vulnerable hosts retrying exploitation. Hosts where ksmbd crash dumps or kernel OOPS files were found prior to patching should be treated as potentially compromised and undergo full memory and filesystem forensic review before being returned to production.

Forensic Artifacts	Kernel crash dumps and OOPS files in '/var/crash/' and '/var/lib/systemd/coredump/' — a successful or attempted exploit of the ksmbd SMB2 buffer length miscalculation flaw may trigger a kernel BUG, NULL pointer dereference, or memory corruption event that generates a crash dump containing the corrupted SMB2 packet structure and kernel stack trace at time of exploitation. | ksmbd kernel log entries from 'journalctl -k | grep ksmbd' and '/var/log/kern.log' — look for anomalous SMB2 NEGOTIATE, SESSION_SETUP, or TREE_CONNECT handling errors, unexpected module unload/reload events, or any error messages referencing buffer length, offset calculation, or memory allocation failures within the ksmbd code path. | Pre-containment TCP 445 network captures (pcap) from tcpdump on the affected Azure Linux 3.0 host's network interface — malformed or oversized SMB2 NEGOTIATE REQUEST packets with anomalous 'StructureSize', 'DialectCount', or 'SecurityBufferLength' field values are the network-level indicator of exploit delivery against this buffer length calculation flaw. | tdnf transaction history and rpm package database state captured before patching ('tdnf history', 'rpm -qa | grep kernel') — establishes the definitive pre-patch exposure window and confirms whether the vulnerable azl3 6.6.130.1-3 package was the active kernel at time of any suspicious SMB activity. | Output of 'ss -tnp sport = :445' and '/proc/net/tcp' captured during the detection phase — documents all active and recently established SMB connections to the ksmbd listener at time of discovery, providing source IP attribution for any pre-patch exploitation attempts that reached an active session state.

Per-Action IR Details

Containment — Identify all Azure Linux 3.0 hosts running the azl3 kernel package at version 6.6.130.1-3. If ksmbd is active and the service is network-reachable, restrict SMB (TCP 445) access at the network perimeter or host firewall to trusted sources only until the patch is applied.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolate affected systems to prevent further exploitation of the unauthenticated network-reachable ksmbd memory corruption vector while preserving operational continuity.

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: On each candidate Azure Linux 3.0 host, immediately insert a host firewall rule using nftables or iptables to block inbound TCP 445 from all sources except known trusted CIDR ranges: 'iptables -I INPUT -p tcp --dport 445 ! -s -j DROP'. At the network perimeter, apply an ACL on the edge router or security group (Azure NSG) blocking inbound TCP 445 to the affected subnet. Document the rule insertion timestamp and approving authority for the incident record. A 2-person team can execute this across a subnet in under 15 minutes using a simple bash loop with ssh.

Evidence: Before inserting firewall rules, capture current nftables/iptables rule state ('iptables-save > /tmp/fw_pre_containment_$(hostname).txt'), active network connections to port 445 ('ss -tnp sport = :445 > /tmp/smb_connections_$(hostname).txt'), and the ksmbd process tree ('ps auxf | grep ksmbd >> /tmp/ksmbd_proc_$(hostname).txt'). These establish the pre-containment exposure baseline and preserve any active attacker sessions for forensic correlation.

Detection — Query your asset inventory and CMDB for Azure Linux 3.0 instances. On candidate hosts, run 'uname -r' to confirm kernel version and 'lsmod | grep ksmbd' or 'systemctl status ksmbd' to determine if ksmbd is loaded and running. Review network flow logs for unexpected SMB traffic (TCP/445) inbound to Azure Linux hosts.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate kernel version telemetry, ksmbd service state, and inbound SMB flow data to determine which hosts are exposed and whether pre-patch exploitation attempts have already occurred.

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: Run a one-liner inventory sweep across all Linux hosts via SSH: 'for h in $(cat hosts.txt); do ssh $h "uname -r; lsmod | grep ksmbd; systemctl is-active ksmbd" 2>/dev/null; done'. For network flow detection without a SIEM, use 'tcpdump -i any -nn port 445 -w /tmp/smb_capture_$(hostname)_$(date +%s).pcap' running for 30 minutes on each exposed host to capture inbound SMB negotiation attempts. Parse ksmbd kernel log output from '/var/log/kern.log' or 'journalctl -k | grep ksmbd' to identify any SMB2 NEGOTIATE or SESSION_SETUP anomalies that may indicate exploit probing against the buffer length calculation flaw. Deploy osquery with a query against 'kernel_info' and 'kernel_modules' tables for scalable detection across a fleet.

Evidence: Collect '/var/log/kern.log' and 'journalctl -k --since -72h' output filtered for 'ksmbd' to capture any kernel warnings, NULL pointer dereferences, or BUG/OOPS entries that would indicate the buffer length miscalculation was triggered. Capture NetFlow or 'tcpdump' data showing the source IPs, connection frequency, and SMB dialect negotiation patterns for all inbound TCP 445 connections to Azure Linux 3.0 hosts — unexpected SMB2 NEGOTIATE requests from external or non-standard internal IPs are the primary pre-exploitation indicator for this unauthenticated attack vector.

Eradication — Apply the patched azl3 kernel package via the standard Azure Linux package manager (tdnf update kernel) or through your Azure Linux patching pipeline. Confirm the updated package version supersedes 6.6.130.1-3. Reference the MSRC April 2026 update guide for the specific fixed build identifier.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the vulnerable azl3 kernel 6.6.130.1-3 from all affected hosts by deploying the MSRC-designated fixed package, eliminating the ksmbd SMB2 buffer length flaw from the environment.

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: Run 'tdnf update kernel -y && tdnf verify kernel' on each affected host and capture the output to a timestamped log: 'tdnf update kernel -y 2>&1 | tee /tmp/patch_$(hostname)_$(date +%s).log'. Verify the installed version with 'rpm -q kernel' and confirm it exceeds azl3 6.6.130.1-3. For a 2-person team managing multiple hosts, orchestrate via a simple SSH loop or Ansible ad-hoc command: 'ansible -i inventory azl3_hosts -m command -a "tdnf update kernel -y" --become'. Cross-reference the installed build number against the MSRC April 2026 advisory fixed build identifier before marking eradication complete.

Evidence: Before patching, capture the pre-patch kernel package state: 'rpm -qi kernel > /tmp/kernel_pre_patch_$(hostname).txt' and 'tdnf history > /tmp/tdnf_history_pre_patch_$(hostname).txt'. Preserve any ksmbd-related crash dumps or kernel OOPS files from '/var/crash/' or '/var/lib/systemd/coredump/' — these would contain memory state evidence of whether the SMB2 buffer length vulnerability was triggered prior to patching and are critical for determining if exploitation occurred before remediation.

Recovery — After patching, reboot affected hosts to load the updated kernel. Re-verify kernel version with 'uname -r'. Confirm ksmbd behavior is normal if the service is required. Re-enable any SMB firewall rules that were restricted during containment. Monitor SMB-facing logs for anomalous connection attempts for 72 hours post-patch.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore Azure Linux 3.0 hosts to verified operational state under the fixed kernel, confirm ksmbd service integrity post-reboot, and re-enable controlled SMB access with active monitoring to detect any post-patch exploitation attempts.

Controls: NIST IR-4 (Incident Handling), NIST SI-6 (Security and Privacy Function Verification), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 8.2 (Collect Audit Logs)

Compensating: Post-reboot, run the following verification sequence: 'uname -r' (confirm fixed kernel is loaded), 'lsmod | grep ksmbd' (confirm module state matches expected), 'systemctl status ksmbd' (confirm service health), and

'journalctl -k -b | grep -i ksmbd' (confirm no error or BUG output on fresh boot). Re-enable firewall rules using 'iptables-restore < /tmp/fw_pre_containment_$(hostname).txt' only after kernel version is confirmed. For 72-hour monitoring without SIEM, run a continuous 'tcpdump -i any -nn port 445' capture with rolling 15-minute files ('tcpdump -i any -nn port 445 -G 900 -w /tmp/smb_monitor_%Y%m%d_%H%M%S.pcap') and review for anomalous SMB2 NEGOTIATE requests from unexpected sources.

Evidence: Capture post-reboot kernel version confirmation and ksmbd service status to a timestamped recovery verification log. Preserve the 72-hour post-patch SMB network captures for 30 days — any SMB2 session attempts from previously observed suspicious source IPs during this window would indicate an active threat actor retrying the exploit vector and require immediate re-escalation. Collect 'journalctl -k --since reboot | grep ksmbd' output as the baseline integrity record for the patched host.

Post-Incident — Assess whether ksmbd is operationally required on all affected hosts; disable the service where it is not needed to reduce attack surface. Review kernel patch SLA adherence for CVSS Critical findings. Evaluate whether Azure Linux 3.0 hosts with network-facing kernel services are covered by your vulnerability scanning and detection rule inventory.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: document lessons learned regarding ksmbd exposure, SLA gaps for CVSS Critical kernel patches, and coverage gaps in vulnerability scanning for Azure Linux 3.0 kernel-level services to prevent recurrence.

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-2 (Flaw Remediation), NIST RA-3 (Risk Assessment), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Use 'systemctl disable ksmbd --now' on all hosts where SMB file sharing is not operationally required, and verify with 'systemctl is-enabled ksmbd'. Document disabled hosts in CMDB. For SLA review, extract patch deployment timestamps from tdnf history logs ('tdnf history | grep kernel') across all affected hosts and measure against your Critical patch SLA window. Write a Sigma rule targeting Linux syslog for ksmbd kernel module load events and deploy it to your log aggregator to ensure future ksmbd activation on non-SMB hosts generates an alert. Add Azure Linux 3.0 kernel package versioning to your next vulnerability scan template using OpenSCAP or a custom osquery scheduled query against the 'rpm_packages' table.

Evidence: Compile the full incident timeline: initial detection timestamp, containment action timestamps, patch deployment timestamps, and reboot timestamps per host. Document any hosts where ksmbd was found running but not in the CMDB or asset inventory — these represent asset visibility gaps that require remediation. Retain all collected SMB network captures, kernel logs, and patch verification logs for a minimum of 90 days to support any downstream forensic or compliance review triggered by this CVSS 9.8 finding.

Detection Guidance

Primary detection approach: asset inventory query for Azure Linux 3.0 hosts running azl3 kernel 6.6.130.1-3. On-host: run 'uname -r' to confirm affected kernel version; run 'lsmod | grep ksmbd' or check 'systemctl status ksmbd' to confirm service state. Network: query firewall and flow logs for inbound TCP/445 connections to Azure Linux hosts, unexpected external sources warrant investigation. SIEM query example (adapt to your log schema; assumes Splunk-compatible field naming): filter events where dest_port=445 AND dest_host in [Azure Linux 3.0 asset group] AND src_ip not in [approved SMB source list]. No public exploit code or IOCs confirmed at time of disclosure; behavioral detection should focus on SMB session anomalies and memory fault indicators (kernel oops, segfault logs) on affected hosts.

Framework Mappings

MITRE-ATTACK

- **T1210** — Exploitation of Remote Services

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-16** — Memory Protection
- **SI-10** — Information Input Validation

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
T1210	Exploitation of Remote Services	Lateral-Movement

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31478	T1
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Apr	T1
CVE-2026-31478 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-31478	T1
CVE-2026-31478	https://access.redhat.com/security/cve/cve-2026-31478	T3
CVE Record: CVE-2026-31478	https://www.cve.org/CVERecord?id=CVE-2026-31478	T3

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