INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-01 07:12 UTC

smb: server: avoid double-free in smb_direct_free_sendmsg after smb_direct_flush_send_list()

CVE VULNERABILITY | **CRITICAL** | CVSS 9.8

SCC Item ID	SCC-CVE-2026-0109
Type	CVE Vulnerability
CVE ID	CVE-2026-31608
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	2026-05-01T01:39:32
Discovery Source	Msrc Patch Tuesday

Executive Summary

A critical memory corruption flaw in the Linux kernel's SMB server component affects Microsoft's Azure Linux 3.0 platform, specifically the azl3 kernel package version 6.6.130.1-3. Organizations running workloads on Azure Linux 3.0 with SMB Direct (RDMA-based transport) enabled are exposed to potential remote code execution or service disruption without authentication. Unpatched systems hosting file shares or storage workloads over SMB Direct represent a direct operational risk.

Technical Analysis

CVE-2026-31608 is a double-free memory corruption vulnerability (CWE-415) in the Linux kernel SMB server implementation. The defect exists in smb_direct_free_sendmsg() when invoked after smb_direct_flush_send_list(), allowing a memory allocation to be freed twice. Double-free conditions corrupt heap metadata and can be leveraged to achieve arbitrary code execution or kernel panic. CVSS base score is 9.8 (Critical) with a network attack vector, reflecting no authentication requirement and low complexity. The affected component is the azl3 kernel package version 6.6.130.1-3 on Azure Linux 3.0. SMB Direct uses RDMA transport, meaning exploitation may not traverse standard TCP/IP inspection points. EPSS score is 0.00057 (17.79th percentile), indicating low observed exploitation activity at time of disclosure. The vulnerability maps to MITRE ATT&CK T1210 (Exploitation of Remote Services) and T1499.004 (Application or System Exploitation for DoS). No CISA KEV listing or active threat actor attribution is present. Disclosed as part of Microsoft Patch

Tuesday April 2026. Source: MSRC Update Guide (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31608); NVD (https://nvd.nist.gov/vuln/detail/CVE-2026-31608).

Action Checklist

1. Step 1: Containment, Identify all Azure Linux 3.0 systems running azl3 kernel 6.6.130.1-3. Temporarily restrict SMB Direct (RDMA) access at the network boundary for any internet-facing or high-value systems until the patch is applied. If SMB Direct is not operationally required, disable the smb_direct kernel module: 'echo "install smb_direct /bin/true" >> /etc/modprobe.d/disable-smb-direct.conf'. Reference MSRC advisory at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31608.
2. Step 2: Detection, Query your asset inventory for Azure Linux 3.0 hosts. Confirm kernel version with 'uname -r' or equivalent fleet tooling. Review kernel logs (/var/log/kern.log, dmesg) for heap corruption indicators: KASAN/SLUB error messages referencing smb_direct_free_sendmsg or double-free. Monitor SMB Direct (port 5445) connection logs for unusual connection patterns. No public IOCs or exploit signatures are available at this time.
3. Step 3: Eradication, Apply the updated azl3 kernel package released under Microsoft Patch Tuesday April 2026. Use 'tdnf update kernel' on affected Azure Linux 3.0 hosts or follow the Azure Linux patching workflow documented in Microsoft's update guide. Verify the updated kernel version supersedes 6.6.130.1-3 before rebooting. Confirm patch applicability against MSRC CVRF feed: https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Apr.
4. Step 4: Recovery, After patching and reboot, confirm the running kernel version with 'uname -r'. Re-enable SMB Direct only if required and only after patch validation. Monitor kernel logs and SMB service health for 24-48 hours post-patch. Validate SMB shares and dependent workloads are operating normally. If anomalous behavior occurred prior to patching, treat affected hosts as potentially compromised and conduct memory forensics before returning them to production.
5. Step 5: Post-Incident, Review whether SMB Direct is enabled by default on Azure Linux 3.0 builds in your environment and whether it is operationally required. Implement a control to alert on kernel module loads for smb_direct on hosts where it is not approved. Assess patch deployment SLA against your critical vulnerability policy; CVSS 9.8 should trigger an emergency patch cycle. Add Azure Linux kernel packages to your Patch Tuesday monitoring scope.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and initiate breach assessment if: (1) KASAN double-free messages referencing smb_direct_free_sendmsg are found in kernel logs prior to patching, indicating the vulnerability was triggered — whether by exploit or bug — on a production host; (2) SMB Direct port 5445 connection logs show inbound connections from external or unexpected internal IPs to unpatched Azure Linux 3.0 hosts during the exposure window, suggesting active exploitation attempts against a CVSS 9.8 unauthenticated RCE; or (3) memory forensics on a suspect host reveals kernel heap artifacts inconsistent with normal smb_direct operation, which may indicate successful code execution requiring regulatory breach notification if PII or PHI workloads were hosted on the affected SMB shares.

Recovery Notes	After applying the updated azl3 kernel package and rebooting, verify the running kernel version supersedes 6.6.130.1-3 via 'uname -r' before re-enabling SMB Direct on any host. Monitor /var/log/kern.log and journalctl -k continuously for 48 hours post-patch for any residual KASAN or SLUB error messages in the smb_direct code path, which would indicate either patch misapplication or a separate memory corruption condition. Any host that logged KASAN double-free events referencing smb_direct_free_sendmsg prior to patching must undergo LiME-based memory forensics and be treated as potentially compromised before returning to production, as successful exploitation of this CVSS 9.8 flaw could have achieved kernel-level code execution without authentication.
Forensic Artifacts	KASAN/SLUB double-free kernel log entries: grep output from /var/log/kern.log and journalctl -k for strings matching 'KASAN', 'double free', 'smb_direct_free_sendmsg', 'smb_direct_flush_send_list', or 'use-after-free' — these are the direct kernel-level evidence that the CVE-2026-31608 memory corruption path was reached on the affected azl3 host | RDMA/SMB Direct network connection records: tcpdump capture on port 5445 and 'ss -tnp sport = :5445' snapshots timestamped before containment, plus Azure NSG flow logs for the affected VMs — used to identify external IPs that established or attempted SMB Direct connections to unpatched hosts during the exposure window | Kernel heap memory image (LiME dump): full physical memory capture from the running azl3 6.6.130.1-3 kernel before reboot, analyzed with Volatility3 to inspect smb_direct-related slab cache allocations (struct smb_direct_sendmsg) for signs of heap manipulation, shellcode, or modified function pointers consistent with kernel RCE exploitation via the double-free path | RPM package database state: 'rpm -qa | grep kernel' and 'rpm -V kernel' output captured at detection time to establish exact vulnerable package version (azl3 6.6.130.1-3) and verify package integrity was not further tampered with on affected hosts | lsmod and /proc/modules snapshots: captured at containment time to confirm whether smb_direct was loaded and active, and to identify any co-loaded modules (e.g., rdma_cm, ib_core) that form the RDMA stack — relevant because a kernel RCE via this vulnerability could load or modify kernel modules post-exploitation, making the module list a secondary indicator of compromise

Per-Action IR Details

Step 1: Containment — Identify all Azure Linux 3.0 systems running azl3 kernel 6.6.130.1-3. Temporarily restrict SMB Direct (RDMA) access at the network boundary for any internet-facing or high-value systems until the patch is applied. If SMB Direct is not operationally required, disable the smb_direct kernel module: 'echo "install smb_direct /bin/true" >> /etc/modprobe.d/disable-smb-direct.conf'. Reference MSRC advisory at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31608.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: short-term containment to limit immediate impact while preserving evidence and system availability for forensic analysis

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

Compensating: For teams without enterprise network ACL tooling: apply a host-based iptables/nftables rule to drop inbound TCP/UDP port 5445 (SMB Direct over RDMA) immediately — 'iptables -I INPUT -p tcp --dport 5445 -j DROP && iptables -I INPUT -p udp --dport 5445 -j DROP'. Enumerate all Azure Linux 3.0 hosts via Azure CLI: 'az vm list --query "[?storageProfile.osDisk.osType=='Linux'].[name,resourceGroup]" -o table' then cross-reference kernel version remotely using 'az vm run-command invoke --command-id RunShellScript --scripts "uname -r"'. Module blacklisting via /etc/modprobe.d/disable-smb-direct.conf requires a reload — confirm the module is not currently loaded with 'lsmod \| grep smb_direct' before assuming containment is effective.

Evidence: Before blacklisting the smb_direct module or dropping firewall rules, capture: (1) current lsmod output ('lsmod \| grep smb' saved to file with timestamp) to document whether smb_direct was loaded at containment time; (2)

active RDMA connection state via 'rdma stat' or 'ss -tnp | grep 5445' to identify any sessions active at the moment of containment; (3) a full dmesg dump ('dmesg -T > /tmp/dmesg_pre_containment_$(date +%s).txt') to preserve any KASAN/SLUB double-free messages referencing smb_direct_free_sendmsg or smb_direct_flush_send_list that may exist prior to module removal; (4) /proc/net/dev and /proc/net/tcp snapshots to record network state before ACL changes alter visibility.

Step 2: Detection — Query your asset inventory for Azure Linux 3.0 hosts. Confirm kernel version with 'uname -r' or equivalent fleet tooling. Review kernel logs (/var/log/kern.log, dmesg) for heap corruption indicators: KASAN/SLUB error messages referencing smb_direct_free_sendmsg or double-free. Monitor SMB Direct (port 5445) connection logs for unusual connection patterns. No public IOCs or exploit signatures are available at this time.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate kernel-level memory corruption indicators with network-layer anomalies to assess whether exploitation attempts have occurred against azl3 kernel 6.6.130.1-3

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-3 (Content of Audit Records), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without a SIEM, run the following on each candidate host to surface KASAN/SLUB double-free indicators specific to this vulnerability: 'grep -E "KASAN|double.free|smb_direct_free_sendmsg|smb_direct_flush_send_list|SLUB.*smb" /var/log/kern.log /var/log/syslog 2>/dev/null | tee /tmp/cve_2026_31608_kernlog_$(hostname)_$(date +%s).txt'. For fleet-scale detection without EDR, deploy a one-line osquery query: "SELECT * FROM kernel_panics WHERE message LIKE '%smb_direct%' OR message LIKE '%double free%';" or use 'journalctl -k --since="7 days ago" | grep -iE "smb_direct|double.free|KASAN|use-after-free"'. Capture port 5445 traffic with 'tcpdump -i any port 5445 -w /tmp/smb_direct_$(date +%s).pcap' for retrospective analysis in Wireshark, filtering for malformed SMB Direct negotiation frames that could indicate exploit probing.

Evidence: Forensic evidence to collect before any remediation action: (1) full kernel log excerpt from /var/log/kern.log and journalctl -k output covering the past 7-30 days, specifically preserving any KASAN BUG reports, SLUB corruption messages, or stack traces that name smb_direct_free_sendmsg or smb_direct_flush_send_list — these are the direct kernel-level artifacts of a double-free trigger against CVE-2026-31608; (2) /var/log/samba/ or ksmbd debug logs (if ksmbd is the SMB server component in use on Azure Linux 3.0) for anomalous session teardown sequences that could precede the double-free condition in smb_direct_flush_send_list; (3) netstat/ss snapshots showing historical RDMA port 5445 connection counts and source IPs ('ss -tnp sport = :5445' output timestamped); (4) /proc/slabinfo captured at detection time to show slab allocator state, which may reflect corruption artifacts in smb_direct-related caches; (5) Azure platform-level NSG flow logs from Azure Monitor for inbound port 5445 connections to identify external sources attempting SMB Direct connections.

Step 3: Eradication — Apply the updated azl3 kernel package released under Microsoft Patch Tuesday April 2026. Use 'tdnf update kernel' on affected Azure Linux 3.0 hosts or follow the Azure Linux patching workflow documented in Microsoft's update guide. Verify the updated kernel version supersedes 6.6.130.1-3 before rebooting. Confirm patch applicability against MSRC CVRF feed: https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Apr.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the root cause of the incident by applying the vendor-supplied kernel patch that corrects the double-free memory corruption in smb_direct_free_sendmsg, eliminating the exploitable code path in the azl3 6.6.130.1-3 package

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

Compensating: For environments without automated patch management (no WSUS/Ansible/Azure Update Manager): execute manually on each Azure Linux 3.0 host — 'tdnf clean all && tdnf makecache && tdnf update kernel -y && rpm

-q kernel | sort -V | tail -1' — and verify the installed version supersedes 6.6.130.1-3 before scheduling reboot. Script this across a fleet using Azure CLI run-command: 'az vm run-command invoke -g -n --command-id RunShellScript --scripts "tdnf update kernel -y && uname -r"'. Before rebooting, use 'rpm -V kernel' (RPM verify) to confirm package integrity against the signed RPM database, validating the patch was not tampered with during delivery — this addresses NIST SI-7 (Software, Firmware, and Information Integrity) without enterprise tooling.

Evidence: Before executing 'tdnf update kernel', preserve the pre-patch system state as forensic baseline: (1) capture 'rpm -qa | grep kernel' output to document the exact vulnerable package version (azl3 kernel 6.6.130.1-3) as evidence of exposure; (2) collect a full memory image if exploitation is suspected — use 'avml' (Acquire Volatile Memory for Linux) to dump /proc/kcore or use LiME kernel module to capture RAM before the reboot destroys volatile memory containing any post-exploitation artifacts (shellcode, ROP chains, injected code) that a CVSS 9.8 RCE exploit against this double-free could have placed in kernel heap space; (3) preserve /proc/modules output showing the smb_direct module load state and all dependent modules at patch time; (4) copy /boot/vmlinuz and /boot/config- for the vulnerable kernel to offline storage for potential later analysis.

Step 4: Recovery — After patching and reboot, confirm the running kernel version with 'uname -r'. Re-enable SMB Direct only if required and only after patch validation. Monitor kernel logs and SMB service health for 24-48 hours post-patch. Validate SMB shares and dependent workloads are operating normally. If anomalous behavior occurred prior to patching, treat affected hosts as potentially compromised and conduct memory forensics before returning them to production.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore Azure Linux 3.0 SMB Direct services to a known-good state using the patched azl3 kernel, verify integrity of the memory management subsystem, and maintain heightened monitoring of smb_direct module behavior before declaring systems safe for production workloads

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 7.2 (Establish and Maintain a Remediation Process)

Compensating: Verify patch effectiveness without enterprise monitoring tooling: (1) confirm patched kernel is running — 'uname -r' must show a version string superseding 6.6.130.1-3; (2) confirm smb_direct module version — 'modinfo smb_direct | grep version'; (3) for 24-48 hour post-patch monitoring without a SIEM, run a cron job every 15 minutes: '*/15 * * * * root grep -E "KASAN|double.free|smb_direct_free_sendmsg" /var/log/kern.log >> /var/log/cve_2026_31608_postpatch_monitor.log 2>&1' — any hits after the patched kernel is running indicate either patch failure or a separate memory corruption issue requiring escalation; (4) for hosts where pre-patch anomalous behavior was observed, perform memory forensics using LiME + Volatility3 before returning to production, specifically examining smb_direct-related kernel heap allocations for signs of post-exploitation persistence (e.g., modified function pointers in the smb_direct send path).

Evidence: Before re-enabling SMB Direct or returning hosts to production, collect: (1) 'uname -r' output and 'rpm -q kernel' saved with timestamp as the authoritative post-patch version attestation; (2) dmesg output immediately post-reboot ('dmesg -T | head -500') to confirm clean kernel initialization without KASAN or SLUB errors in the SMB Direct initialization path; (3) if pre-patch anomalous behavior was logged, capture a post-patch LiME memory image for comparison against pre-patch baseline — look for residual kernel heap artifacts in smb_direct slab caches (struct smb_direct_sendmsg allocations) that should be clean after reboot but could indicate prior exploitation modified persistent kernel structures; (4) SMB service functional validation — capture output of 'smbstatus' and 'ksmbd.mountd --version' (if applicable) showing clean session establishment after patch.

Step 5: Post-Incident — Review whether SMB Direct is enabled by default on Azure Linux 3.0 builds in your environment and whether it is operationally required. Implement a control to alert on kernel module loads for smb_direct on hosts where it is not approved. Assess patch deployment SLA against your critical vulnerability policy; CVSS 9.8 should trigger an emergency patch cycle. Add Azure Linux kernel packages to your Patch Tuesday monitoring scope.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned review to determine why azl3 kernel 6.6.130.1-3 with smb_direct enabled was present in the environment, update patch SLA policy to reflect CVSS 9.8 emergency timelines, and implement permanent detection controls for unauthorized smb_direct module loading

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality), NIST AU-12 (Audit Record Generation), 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), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Implement a permanent, free detection control for unauthorized smb_direct loading: add a udev rule or auditd rule to alert on module loads — 'auditctl -w /sys/module/smb_direct -p rwxa -k smb_direct_load' — then monitor /var/log/audit/audit.log for key 'smb_direct_load'. For Patch Tuesday monitoring of Azure Linux kernel packages without a commercial vulnerability management platform, subscribe to the Azure Linux GitHub release feed (https://github.com/microsoft/azurelinux/releases) via RSS and configure a cron-driven script that queries 'tdnf check-update kernel 2>/dev/null | grep kernel' weekly and emails the security team if a kernel update is available. Write a YARA rule targeting the vulnerable smb_direct_free_sendmsg/smb_direct_flush_send_list symbol pattern in kernel module files for use with ClamAV or manual scanning of new kernel packages before deployment.

Evidence: Post-incident documentation artifacts to retain: (1) the pre-patch 'rpm -qa | grep kernel' output from all affected hosts as the official record of vulnerability exposure scope; (2) the timeline of KASAN/SLUB error messages from kern.log across all affected systems, establishing whether exploitation was attempted or successful during the exposure window; (3) the full dmesg and kernel log archive from affected hosts for the period when azl3 6.6.130.1-3 was running — this is the primary forensic record for determining if CVE-2026-31608 was triggered in your environment; (4) network flow logs for port 5445 covering the exposure window, preserved per NIST AU-11 (Audit Record Retention) requirements, to support any future breach notification assessment if RCE exploitation is later confirmed by threat intelligence.

Detection Guidance

No public exploit code or IOCs are available as of disclosure. Detection is primarily asset-based and log-based. (1) Asset detection: identify hosts running 'uname -r' output matching azl3 kernel 6.6.130.1-3 via endpoint management tooling or cloud inventory APIs. (2) Kernel log analysis: search dmesg and /var/log/kern.log for strings including 'double free', 'KASAN', 'smb_direct', or 'use-after-free'. If KASAN is enabled in the kernel build, reports will include the function name smb_direct_free_sendmsg; however, production Azure Linux kernels may not include KASAN by default. Check your kernel configuration. (3) Network-level: monitor for unexpected SMB Direct traffic on TCP/UDP 5445 from external sources. SMB Direct over RDMA may bypass standard SMB (port 445) monitoring; ensure RDMA-capable network adapters are inventoried. (4) Crash telemetry: kernel panics (oops logs) referencing smb_direct should be triaged as potential exploitation attempts. Behavioral detection for heap corruption exploitation is difficult without kernel-level instrumentation; prioritize patching over detection reliance.

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

CIS-V8

- **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
T1210	Exploitation of Remote Services	Lateral-Movement

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-31608	T1
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Apr	T1
CVE-2026-31608 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-31608	T1
CVE-2026-31608 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-31608	T3
CVE-2026-31608 - Vulnerability Details - OpenCVE	https://app.opencve.io/cve/CVE-2026-31608	T3

Generated 2026-05-01 07:12 UTC by TJS Security Command Center