INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-11 07:43 UTC

Apache HTTP Server Heap Underflow via Signed Char Overflow in ap_regname (CVE-2026-44631)

CVE VULNERABILITY | CRITICAL | CVSS 9.4

SCC Item ID	SCC-CVE-2026-0291
Type	CVE Vulnerability
CVE ID	CVE-2026-44631
Severity	CRITICAL
CVSS Base Score	9.4
EPSS Score	0.0004 (13th percentile)
Affected Products	Microsoft azl3 httpd 2.4.67-1 on Azure Linux 3.0
Published	2026-06-11T01:44:37
Discovery Source	Msrc Patch Tuesday

Executive Summary

A critical heap underflow vulnerability (CVE-2026-44631, CVSS 9.4) was disclosed in Apache HTTP Server (version azl3 httpd 2.4.67-1, packaged for Microsoft Azure Linux 3.0) as part of the June 2026 Patch Tuesday cycle. An unauthenticated attacker who can send crafted HTTP requests to an affected server may trigger heap memory corruption, potentially achieving arbitrary code execution or causing a denial of service. Organizations running this specific package on Azure Linux 3.0 should treat patching as an immediate priority, as a successful exploit against an internet-facing web server can result in full system compromise.

Technical Analysis

CVE-2026-44631 is a heap underflow vulnerability in the Apache HTTP Server ap_regname function, caused by a signed char overflow (CWE-191: Integer Underflow; CWE-122: Heap-based Buffer Overflow, analyst-inferred based on the described vulnerability class, not vendor-assigned in the official CVE record). The overflow corrupts heap memory in a way that may allow arbitrary code execution or denial of service. The affected package is Microsoft's Azure Linux 3.0 build: azl3 httpd 2.4.67-1. CVSS base score is 9.4 (Critical). EPSS score is 0.00043 (13.4th percentile) as of disclosure, indicating low observed exploitation activity at this time. The vulnerability maps to MITRE ATT&CK T1190 (Exploit Public-Facing Application). CWE-191 and CWE-122 mappings are analyst-inferred from the described vulnerability class. The vulnerability is not listed in the CISA KEV catalog. No public proof-of-concept or active exploitation has been confirmed. Sources: MSRC Update

Guide (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-44631), NVD (https://nvd.nist.gov/vuln/detail/CVE-2026-44631).

Action Checklist

1. Step 1: Identification. Identify all Azure Linux 3.0 hosts running azl3 httpd 2.4.67-1. If internet-facing and unpatched, place a WAF or reverse proxy in front to filter malformed HTTP requests while the patch is applied. Restrict direct external access to affected servers until remediation is confirmed.
2. Step 2: Detection. Query asset inventory and configuration management tooling for azl3 httpd 2.4.67-1 on Azure Linux 3.0 (CIS 1.1, CIS 2.1). Review Apache access and error logs for anomalous requests to endpoints that invoke ap_regname processing, look for malformed request paths, unexpected 500-series errors, or process crashes in /var/log/httpd/error_log. Enable core dump analysis if crashes are observed. No confirmed IOC signatures are available at this time given low EPSS and no active exploitation reports. Monitor MSRC (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-44631) and NVD for updated IOC or exploitation data.
3. Step 3: Eradication. Apply the Microsoft-provided patch for CVE-2026-44631 via the Azure Linux 3.0 package manager (tdnf update httpd or equivalent). Confirm the updated package version resolves the azl3 httpd 2.4.67-1 build. Reference the MSRC Update Guide for the specific package version that addresses the vulnerability. After patching, validate the running httpd version to confirm the update was applied (httpd -v).
4. Step 4: Recovery. Restart the httpd service and verify it returns to normal operation. Monitor Apache error logs and system logs for residual crashes or anomalous heap behavior post-patch (NIST AU-6, CIS 8.2). Confirm WAF or proxy rules applied during containment remain in place or are reviewed for retention. Run a vulnerability scan against patched hosts to verify the CVE is no longer flagged.
5. Step 5: Post-Incident Activities. Document the time between disclosure (June 2026 Patch Tuesday) and patch application as a patching SLA metric. Evaluate whether automated patch management is in place for Azure Linux 3.0 hosts (CIS 7.3, CIS 7.4). Review whether internet-facing Apache instances have a WAF as a standard architectural control (NIST AC-4). Assess asset inventory completeness, were all affected hosts identified promptly (CIS 1.1, CIS 2.1)?

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to senior IR leadership and Azure infrastructure ownership if any Azure Linux 3.0 host running azl3 httpd 2.4.67-1 shows evidence of successful exploitation — specifically, unexpected child process crashes correlated with inbound HTTP requests containing crafted path data, presence of anomalous processes spawned by httpd (indicating potential code execution), or any outbound connections from httpd worker processes to non-standard destinations — or if the organization cannot achieve patch application within 72 hours for internet-facing instances given the CVSS 9.4 unauthenticated attack vector.

Recovery Notes	After patching, maintain heightened monitoring of `/var/log/httpd/error_log` and `/var/log/messages` on all formerly vulnerable hosts for a minimum of 7 days post-patch, specifically watching for late-appearing crash signatures (`SIGSEGV`, `SIGABRT`, heap corruption messages) that could indicate pre-patch exploitation that was not immediately apparent or a patch regression. Verify that the WAF or Nginx reverse proxy rules deployed during containment remain active and review whether they should be retained permanently as a defense-in-depth architectural control for all internet-facing Apache instances, regardless of patch status. Run a follow-up asset inventory query 30 days post-incident to confirm no new Azure Linux 3.0 hosts have been deployed with the vulnerable azl3 httpd 2.4.67-1 package.
Forensic Artifacts	/var/log/httpd/error_log — contains child process crash records (SIGSEGV, SIGABRT, 'heap corruption detected') timestamped to the moment ap_regname triggered the signed char overflow; the most direct indicator of exploitation attempts against CVE-2026-44631 | /var/log/httpd/access_log — the HTTP request record immediately preceding each 500-series error or crash entry in error_log; will contain the crafted path or header value used to trigger the heap underflow in ap_regname, identifiable by anomalous path length, binary characters, or malformed URI encoding | Core dump files at the path specified in /proc/sys/kernel/core_pattern — generated when httpd worker processes abort due to heap corruption; analyzable with GDB to confirm the faulting instruction is within the ap_regname code path and to reconstruct the heap state at time of crash | /proc//maps and /proc//smaps (captured before any service restart) — documents the heap segment layout of the vulnerable httpd process, enabling identification of whether heap metadata was overwritten consistent with a heap underflow exploitation attempt | RPM transaction log at /var/log/dnf.rpm.log or /var/log/tdnf.log — records the pre-patch presence of azl3 httpd 2.4.67-1 and the post-patch replacement package version, serving as the authoritative forensic record of when the vulnerable package was installed and when it was remediated

Per-Action IR Details

Step 1: Containment — Identify all Azure Linux 3.0 hosts running azl3 httpd 2.4.67-1. If internet-facing and unpatched, place a WAF or reverse proxy in front to filter malformed HTTP requests while the patch is applied. Restrict direct external access to affected servers until remediation is confirmed.

NIST Phase: Containment

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

Controls: NIST AC-4 (Information Flow Enforcement), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: Run `tdnf list installed | grep httpd` across Azure Linux 3.0 hosts via a parallel SSH loop or Azure Run Command to enumerate all instances of azl3 httpd 2.4.67-1. For WAF-equivalent filtering without enterprise tooling, deploy an Nginx reverse proxy with a custom `location` block that rejects requests containing null bytes, oversized path segments, or malformed URI encoding patterns associated with heap underflow triggering — specifically block requests where the path component that would invoke ap_regname parsing exceeds 255 characters or contains binary sequences (`\x00`-`\x1f`). Apply a host-based iptables rule to restrict port 80/443 inbound to known IP ranges until patching is complete.

Evidence: Before isolating, capture a full memory image of any suspected-compromised host using LiME (Linux Memory Extractor) loaded as a kernel module — heap corruption from CVE-2026-44631 may leave traces of the overwritten heap metadata or adjacent chunk headers in memory. Preserve `/var/log/httpd/access_log` and `/var/log/httpd/error_log` as read-only snapshots (`cp --preserve=timestamps`) before any service restart. Document the exact running process state with `ps auxf` and `/proc//maps` to capture the heap layout at time of containment.

Step 2: Detection — Query asset inventory and configuration management tooling for azl3 httpd 2.4.67-1 on Azure Linux 3.0 (CIS 1.1, CIS 2.1). Review Apache access and error logs for anomalous requests to endpoints that invoke ap_regname processing — look for malformed request paths, unexpected 500-series errors, or

process crashes in /var/log/httpd/error_log. Enable core dump analysis if crashes are observed. No confirmed IOC signatures are available at this time given low EPSS and no active exploitation reports. Monitor MSRC (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-44631) and NVD for updated IOC or exploitation data.

NIST Phase: Detection Analysis

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

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-2 (Event Logging)

Compensating: Deploy osquery on Azure Linux 3.0 hosts and run `SELECT name, version, source FROM rpm_packages WHERE name LIKE '%httpd%';` to enumerate all azl3 httpd instances fleet-wide. For log-based detection without a SIEM, run the following bash one-liner against `/var/log/httpd/access_log` to surface requests that may have targeted ap_regname parsing: `grep -E '(GET|POST|HEAD|OPTIONS).{200,}' /var/log/httpd/access_log | grep -E '(500|502|503)'`. Supplement with `grep -i 'segfault\|heap\|corrupted\|abort\|SIGSEGV\|SIGABRT' /var/log/httpd/error_log /var/log/messages` to detect crash indicators. If core dumps are enabled (`/proc/sys/kernel/core_pattern`), collect and analyze with GDB: `gdb /usr/sbin/httpd ` and run `bt full` to identify the faulting frame in the ap_regname call path.

Evidence: The specific exploitation path for CVE-2026-44631 involves a signed char overflow in ap_regname during HTTP request parsing; capture Apache error log entries showing `child pid exit signal Segmentation fault` or `AH00052: child pid exit signal Aborted`, which indicate heap corruption during request processing. Preserve any core dump files generated under the configured core dump path (check `/proc/sys/kernel/core_pattern` on the affected host). Collect `/proc//smaps` and `/proc//status` before any restart to document heap segment sizes. Extract the specific HTTP request that preceded each 500-series error from access logs by correlating timestamps between `access_log` and `error_log` entries — the triggering request will likely contain a crafted path or header value designed to produce a negative signed char value passed to ap_regname.

Step 3: Eradication — Apply the Microsoft-provided patch for CVE-2026-44631 via the Azure Linux 3.0 package manager (tdnf update httpd or equivalent). Confirm the updated package version resolves the azl3 httpd 2.4.67-1 build. Reference the MSRC Update Guide for the specific package version that addresses the vulnerability. After patching, validate the running httpd version to confirm the update was applied (httpd -v).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Run `sudo tdnf update httpd -y` on each Azure Linux 3.0 host and capture the output, specifically verifying that the transaction replaces `azl3 httpd 2.4.67-1` with the patched package version listed in the MSRC advisory for CVE-2026-44631. After update, verify with `rpm -q httpd` to confirm the installed package version matches the fixed build and with `httpd -v` to confirm the running binary reflects the update. For a 2-person team managing multiple hosts, script the update and version check across all affected hosts using Azure Run Command: `az vm run-command invoke --command-id RunShellScript --scripts 'tdnf update httpd -y && rpm -q httpd'`. Cross-reference the output package version string against the MSRC Update Guide entry for CVE-2026-44631 to confirm the specific build that resolves the signed char overflow in ap_regname.

Evidence: Before applying the patch, preserve the pre-patch RPM metadata with `rpm -qi httpd > /tmp/httpd_preupdate.txt` and the binary checksum with `sha256sum /usr/sbin/httpd >> /tmp/httpd_preupdate.txt` — this establishes a forensic baseline confirming the vulnerable build was present and enables post-patch comparison to verify the binary was replaced. If any crashes were observed prior to patching, ensure all core dump files are preserved offline before the patch is applied, as the patch may change the binary and reduce the forensic value of subsequent crash analysis.

Step 4: Recovery — Restart the httpd service and verify it returns to normal operation. Monitor Apache error logs and system logs for residual crashes or anomalous heap behavior post-patch (NIST AU-6, CIS 8.2). Confirm WAF or proxy rules applied during containment remain in place or are reviewed for retention. Run a

vulnerability scan against patched hosts to verify the CVE is no longer flagged.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 8.2 (Collect Audit Logs), NIST AC-4 (Information Flow Enforcement)

Compensating: After `systemctl restart httpd`, immediately tail `/var/log/httpd/error_log` for 15 minutes watching for any recurrence of `SIGSEGV`, `SIGABRT`, or heap corruption messages that would indicate the patch did not fully resolve the vulnerability or a secondary issue exists. Run OpenSCAP or `tdnf check-update` post-patch to confirm no outstanding advisories remain for the httpd package. For vulnerability scan verification without an enterprise scanner, use `rpm -q --changelog httpd | head -20` to confirm the CVE-2026-44631 fix is referenced in the package changelog, and cross-check the installed package version string against the fixed version documented in the MSRC advisory. Retain WAF or Nginx reverse proxy rules applied during containment for a minimum of 30 days as a defense-in-depth control, given the critical CVSS score and unauthenticated attack vector.

Evidence: During the recovery monitoring window, collect a clean snapshot of `/var/log/httpd/error_log` and `/var/log/httpd/access_log` starting from the exact timestamp of the `systemctl restart httpd` command — this establishes a post-patch baseline and captures any residual exploitation attempts against the now-patched service. Preserve the output of `rpm -q httpd` and `sha256sum /usr/sbin/httpd` post-patch as evidence of successful remediation. If the system experienced crashes before patching, verify that `/proc/sys/kernel/core_pattern` is configured to retain post-patch core dumps for at least 72 hours to detect any regression.

Step 5: Post-Incident — Document the time between disclosure (June 2026 Patch Tuesday) and patch application as a patching SLA metric. Evaluate whether automated patch management is in place for Azure Linux 3.0 hosts (CIS 7.3, CIS 7.4). Review whether internet-facing Apache instances have a WAF as a standard architectural control (NIST AC-4). Assess asset inventory completeness — were all affected hosts identified promptly (CIS 1.1, CIS 2.1)?

NIST Phase: Post Incident

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

Controls: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), NIST AC-4 (Information Flow Enforcement), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Produce a lessons-learned document recording: (1) the date CVE-2026-44631 appeared in the June 2026 MSRC Patch Tuesday bulletin, (2) the date each Azure Linux 3.0 host running azl3 httpd 2.4.67-1 was identified via asset inventory, and (3) the date each host was patched — this SLA gap is the primary metric for improving future response to unauthenticated RCE-class vulnerabilities in internet-facing Apache instances. If asset inventory failed to surface all azl3 httpd 2.4.67-1 hosts promptly, implement a recurring osquery scheduled query (`SELECT name, version FROM rpm_packages WHERE name LIKE '%httpd%'`) to maintain a live software inventory for Azure Linux 3.0 hosts. For teams without automated patch management, configure `tdnf-automatic` (the Azure Linux equivalent of `dnf-automatic`) with `apply_updates = yes` for security-classified updates to reduce SLA exposure for future critical CVEs.

Evidence: Retain all collected artifacts from prior steps — pre-patch RPM metadata, core dump files, access and error log snapshots, and the Azure Run Command output confirming patched package versions — as the evidentiary record for this incident. These artifacts collectively document the affected host population, the vulnerability's presence, and the remediation action, fulfilling post-incident reporting requirements and providing the baseline for SLA metric calculation tied to the June 2026 Patch Tuesday disclosure date.

Detection Guidance

No confirmed IOCs (IPs, hashes, domains) are available for CVE-2026-44631 at this time; EPSS of 0.00043 and absence from CISA KEV indicate no observed exploitation activity as of disclosure. Detection focus should be on vulnerable asset identification and anomaly monitoring. Query your asset inventory or CMDB for Azure Linux 3.0 hosts running azl3 httpd 2.4.67-1 (CIS 1.1). Review Apache error logs (/var/log/httpd/error_log) for repeated 500-series errors, segmentation faults, or httpd process crashes, which may indicate heap corruption attempts. Monitor system logs for unexpected process termination of the httpd worker processes. If a WAF is in place, review logs for anomalous or malformed HTTP request patterns targeting the ap_regname code path. Establish a SIEM alert for httpd crash events on Azure Linux 3.0 hosts until patching is complete (NIST AU-6, NIST SI-4). D3FEND countermeasure SFA (System File Analysis) applies: monitor system logs and process state for modification or unexpected termination indicative of heap corruption. D3FEND countermeasure PBWSAM (Proxy-based Web Server Access Mediation) applies as a detection and containment layer.

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application

NIST-800-53R5

- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity

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
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-44631	T1

Source	URL	Tier
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Jun	T1
CVE-2026-44631 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-44631	T1
CVE-2026-44631 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-44631	T3
CVE-2026-44631 Common Vulnerabilities and Exposures | SUSE	https://www.suse.com/security/cve/CVE-2026-44631.html	T3

Generated 2026-06-11 07:43 UTC by TJS Security Command Center