INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-30 14:08 UTC

CVE-2026-32211: Azure MCP Server Critical Information Disclosure Vulnerability

CVE VULNERABILITY | CRITICAL | CVSS 9.1

SCC Item ID	SCC-CVE-2026-0101
Type	CVE Vulnerability
CVE ID	CVE-2026-32211
Severity	CRITICAL
CVSS Base Score	9.1
EPSS Score	0.0008 (23th percentile)
Affected Products	Microsoft Azure Web Apps (Azure MCP Server component)
Published	2026-04-30T07:00:00
Discovery Source	Msrc Patch Tuesday

Executive Summary

Microsoft disclosed CVE-2026-32211, a critical information disclosure vulnerability (CVSS 9.1) in the Azure MCP Server component of Azure Web Apps, as part of April 2026 Patch Tuesday. The flaw allows unauthorized parties to access sensitive information hosted within affected Azure environments. Organizations running Azure Web Apps with the MCP Server component should treat patching as a priority action, given the critical severity rating and the sensitive nature of data typically processed by AI-integrated cloud services.

Technical Analysis

CVE-2026-32211 affects the Model Context Protocol (MCP) Server component within Microsoft Azure Web Apps. CVSS base score is 9.1 (Critical). CVSS vector string and CWE classification were not available in the accessible source data at time of publication; these should be confirmed against the authoritative MSRC advisory and NVD record before final distribution. MITRE ATT&CK technique T1530 (Data from Cloud Storage) is mapped to this vulnerability, indicating the disclosure path likely involves unauthorized read access to cloud-resident data. EPSS score is 0.00078 (22.9th percentile), suggesting low observed exploitation probability at time of disclosure, though this metric is early-stage and should not override the CVSS severity for patching decisions. The vulnerability is not currently listed on the CISA Known Exploited Vulnerabilities catalog. Specific affected version ranges, exploitation preconditions, and the precise disclosure mechanism were not independently verified beyond MSRC advisory metadata. Source quality score: 0.856. Primary references:

MSRC Update Guide and NVD.

Action Checklist

1. Step 1: Containment, Identify all Azure Web Apps instances running the MCP Server component across your tenant. Restrict public access to affected endpoints via Azure App Service access restrictions or front-end WAF rules until patching is confirmed. Check the MSRC advisory at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-32211 for any available workaround guidance.
2. Step 2: Detection, Review Azure Activity Logs and App Service diagnostic logs for anomalous GET or data-read requests against MCP Server endpoints, particularly from unexpected IP ranges or unauthenticated identities. Cross-reference with Microsoft Defender for Cloud alerts mapped to T1530 (Data from Cloud Storage). Query Azure Monitor for unusual data egress volumes from affected Web App resources.
3. Step 3: Eradication, Apply the Microsoft-issued patch as documented in the April 2026 MSRC Update Guide (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-32211). Note: At time of publication, specific patch ID and updated runtime version details were not available in the source dataset. Confirm patch applicability and version targeting in the advisory before deployment.
4. Step 4: Recovery, After patching, validate MCP Server endpoints return expected responses without exposing sensitive metadata. Re-enable any access restrictions lifted during assessment. Monitor Defender for Cloud and Azure Monitor for 72 hours post-patch for residual anomalous access patterns.
5. Step 5: Post-Incident, Evaluate whether MCP Server components were deployed with least-privilege data access controls. Review Azure RBAC assignments scoped to Web App identities. If sensitive data was accessible via the MCP component, assess whether a data exposure notification obligation exists under applicable regulatory frameworks.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to CISO and legal counsel if Azure Activity Log or App Service HTTP logs confirm unauthorized unauthenticated GET requests to MCP Server endpoints returning HTTP 200 with non-empty response bodies during the vulnerability exposure window, as this constitutes confirmed data access triggering potential breach notification obligations under applicable data protection regulations; additionally escalate if Azure Monitor egress metrics show anomalous BytesSent volumes from affected Web App resources exceeding 2x the 30-day baseline during that same window.

Recovery Notes	Post-patch recovery for CVE-2026-32211 requires verifying that unauthenticated requests to MCP Server endpoints no longer return the sensitive metadata fields described in the MSRC advisory — do not rely solely on HTTP status codes, as information disclosure vulnerabilities may return 200 with reduced or sanitized payloads rather than 403. Re-enable WAF and access restriction rules to their pre-incident least-privilege configuration and confirm rule order precedence in Azure App Service access restriction lists, as deny rules must be correctly ordered relative to any allow-listed management CIDRs. Maintain elevated monitoring of MCP endpoint traffic and Azure AD token issuance for the Web App managed identity for a minimum of 72 hours post-patch, extending to 7 days if any anomalous access was confirmed during the detection phase.
Forensic Artifacts	Azure App Service HTTP access logs (Kudu console: /home/LogFiles/http/RawLogs/) — filter for GET requests to MCP Server route paths returning HTTP 200 from unauthenticated or unexpected client identities during the vulnerability exposure window; these logs are the primary record of whether the information disclosure was exploited | Azure Monitor AzureDiagnostics table (Log Analytics workspace) — specifically the requestUri_s, clientIP_s, bytesSent_d, csUser_s, and resultCode_d fields for Web App resources hosting the MCP Server component, covering the period from initial MCP Server deployment through patch application | Azure Activity Log export for affected Web App resource IDs — captures control-plane operations including configuration changes, access restriction modifications, and managed identity assignments that may indicate attacker persistence or privilege escalation following initial information disclosure | Azure AD Sign-In Logs for the managed identity or service principal bound to the affected Web App — anomalous token issuance locations, client IPs, or resource access patterns following exploitation of the MCP Server endpoint would appear here as the attacker leverages disclosed credentials or tokens | Azure Network Watcher NSG Flow Logs for the subnet hosting affected Web Apps — captures raw network-layer egress volumes and destination IPs that would reveal data exfiltration attempts following information disclosure, independent of App Service application-layer logging which an attacker might attempt to suppress

Per-Action IR Details

Step 1: Containment — Identify all Azure Web Apps instances running the MCP Server component across your tenant. Restrict public access to affected endpoints via Azure App Service access restrictions or front-end WAF rules until patching is confirmed. Check the MSRC advisory at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-32211 for any available workaround guidance.

NIST Phase: Containment

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

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

Compensating: Use Azure CLI to enumerate all Web Apps with MCP Server: `az webapp list --query "[].{name:name, rg:resourceGroup, state:state}" -o table`, then cross-reference App Service configurations with `az webapp config show --name --resource-group | grep -i mcp`. For access restriction without WAF budget, use `az webapp config access-restriction add --rule-name BlockPublic --action Deny --ip-address 0.0.0.0/0 --priority 100` per affected app. Document each restricted app with timestamp for recovery rollback.

Evidence: Before restricting access, export the current Azure App Service access restriction rules and inbound IP logs via `az webapp log download --name --resource-group `. Capture Azure Activity Log entries scoped to the affected Web App resource IDs showing historical access patterns — run `az monitor activity-log list --resource-id --start-time -o json > activity_baseline.json`. Preserve any existing App Service HTTP logs from `/home/LogFiles/http/RawLogs/` (Kudu console) before WAF rules alter incoming traffic visibility.

Step 2: Detection — Review Azure Activity Logs and App Service diagnostic logs for anomalous GET or data-read requests against MCP Server endpoints, particularly from unexpected IP ranges or unauthenticated identities. Cross-reference with Microsoft Defender for Cloud alerts mapped to T1530 (Data from Cloud Storage). Query Azure Monitor for unusual data egress volumes from affected Web App resources.

NIST Phase: Detection Analysis

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

Controls: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST SI-4 (System Monitoring), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, query Azure Monitor Logs directly via the Azure Portal Log Analytics workspace using KQL: `AzureDiagnostics | where ResourceType == 'SITES' | where requestUri_s contains '/mcp/' | where resultCode_d == 200 | where userAgent_s == '-' or isempty(userAgent_s) | project TimeGenerated, clientIP_s, requestUri_s, bytesSent_d, csUser_s | order by TimeGenerated desc`. For egress anomaly detection without cost tooling, use Azure Cost Management + Azure Network Watcher flow logs exported to a storage account and parsed with a local Python script using `pandas` to flag stddev outliers on `bytesSent_d` per resource per hour. Map findings to MITRE ATT&CK T1530 (Data from Cloud Storage Object) for any confirmed unauthorized reads of MCP-served data.

Evidence: Collect App Service HTTP access logs from the Kudu console (`https://.scm.azurewebsites.net/api/logs/docker`) specifically filtering for GET requests to MCP Server endpoint paths (e.g., `/mcp/`, `/api/mcp/`, or vendor-documented MCP route patterns per MSRC advisory). Export Azure AD sign-in logs for the managed identity or service principal bound to the affected Web App — look for token issuance to unexpected client IPs or geographic anomalies. Pull Azure Network Watcher NSG flow logs for the Web App's hosting subnet covering the 30 days prior to advisory date to identify pre-patch reconnaissance activity consistent with T1530.

Step 3: Eradication — Apply the Microsoft-issued patch as documented in the April 2026 MSRC Update Guide. Confirm the specific patch ID or updated runtime version in the advisory, as version-specific remediation details were not available in the source dataset at time of writing.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST CM-3 (Configuration Change Control), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For teams without automated patch management: use Azure App Service's built-in auto-update for managed runtimes via `az webapp config set --name --resource-group --auto-heal-enabled true` and verify runtime version post-update with `az webapp show --name --resource-group --query 'siteConfig.linuxFxVersion'`. If the MSRC advisory specifies a manual runtime pin, update via `az webapp config set --linux-fx-version '|'`. Document the pre-patch and post-patch runtime version strings as change control evidence. Note: specific patch KB ID and patched runtime version must be confirmed from the live MSRC advisory at https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-32211 before execution — this detail was not available in the source dataset at time of writing.

Evidence: Before applying the patch, capture a snapshot of the current App Service configuration: `az webapp config show --name --resource-group -o json > preatch_config_.json`. Export the current deployment slot settings and any environment variables referencing MCP Server configuration paths (redact secrets before storage). Preserve App Service application logs from the 24 hours preceding patch application as a pre-remediation forensic baseline — these may contain exploit attempt artifacts including malformed MCP request patterns or error traces indicating vulnerability probe activity.

Step 4: Recovery — After patching, validate MCP Server endpoints return expected responses without exposing sensitive metadata. Re-enable any access restrictions lifted during assessment. Monitor Defender for Cloud and Azure Monitor for 72 hours post-patch for residual anomalous access patterns.

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 SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Use `curl -v -H 'Authorization: ' https://.azurewebsites.net/mcp/` (unauthenticated) against the patched MCP Server endpoint and verify the response does NOT contain sensitive metadata fields that the vulnerability exposed — compare response body and headers against pre-patch baseline captures. Re-apply access restrictions via `az webapp config access-restriction remove` for allow-listed CIDRs only. For 72-hour monitoring without Defender for Cloud paid tier, schedule an hourly Azure Monitor KQL query via Logic App (free tier): `AzureDiagnostics | where requestUri_s contains '/mcp/' | where resultCode_d == 200 | summarize count() by clientIP_s, bin(TimeGenerated, 1h)` and alert on new source IPs not seen in the pre-patch baseline.

Evidence: Capture post-patch endpoint response headers and body for MCP Server endpoints as verification evidence — specifically confirm that fields documented in the CVE-2026-32211 advisory as exposed are no longer present in unauthenticated responses. Export a post-patch Azure Activity Log snapshot using the same query used in Step 1 evidence collection to establish a clean-state baseline. Retain the 72-hour post-patch Azure Monitor egress metrics (`az monitor metrics list --resource --metric BytesSent`) for comparison against pre-patch egress anomalies identified in Step 2.

Step 5: Post-Incident — Evaluate whether MCP Server components were deployed with least-privilege data access controls. Review Azure RBAC assignments scoped to Web App identities. If sensitive data was accessible via the MCP component, assess whether a data exposure notification obligation exists under applicable regulatory frameworks.

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 AC-6 (Least Privilege), NIST AU-11 (Audit Record Retention), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 3.2 (Establish and Maintain a Data Inventory)

Compensating: Run `az role assignment list --assignee -o table` to enumerate all RBAC roles granted to the Web App's managed identity — flag any assignments broader than the specific storage account, Key Vault, or data resource the MCP Server legitimately requires. Use `az ad app list --filter 'displayName eq ''''` to identify any application registrations with excessive Microsoft Graph or Azure Resource Manager permissions. For regulatory scoping without a DLP tool, manually cross-reference the data types accessible via MCP endpoints (per App Service environment variables and connection strings: `az webapp config appsettings list --name --resource-group `) against your data inventory to determine PII/PHI/PCI scope of potential exposure.

Evidence: Preserve the complete Azure Activity Log export covering 90 days prior to advisory date for all affected Web App resource IDs — this is the primary forensic record for regulatory breach notification scope assessment and must be retained per NIST AU-11 (Audit Record Retention). Export the Azure AD access reviews and RBAC assignment history for the Web App managed identities as of the vulnerability disclosure date. Retain all artifacts collected in Steps 1-4 (pre-patch configs, HTTP access logs, egress metrics, endpoint response captures) in tamper-evident storage (Azure Blob with immutability policy enabled) for potential regulatory or legal proceedings. Note: regulatory breach notification obligations under GDPR, HIPAA, or CCPA require legal counsel determination — flag this for appropriate authority if sensitive data classes were confirmed accessible.

Detection Guidance

Query Azure Activity Logs for unauthorized or anonymous read operations against Azure Web App resources hosting the MCP Server component. In Microsoft Defender for Cloud, filter alerts for T1530-mapped detections (cloud storage data access anomalies). In Azure Monitor, build a log query targeting AppServiceHTTPLogs or AppServiceConsoleLogs for unexpected 200-series responses on MCP-specific routes from unauthenticated or external principals. Establish a baseline of normal MCP endpoint traffic volume and flag deviations exceeding

two standard deviations. No public IOCs were available in the source data at time of publication; monitor MSRC and threat intelligence feeds for updated indicators.

Framework Mappings

MITRE-ATTACK

- **T1530** — Data from Cloud Storage

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

SOC2-TSC

- **CC6.3** — Authorizes, modifies, or removes access

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1530	Data from Cloud Storage	Collection

Sources

Source	URL	Tier
MSRC Update Guide	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-32211	T1
(consolidated)	https://api.msrc.microsoft.com/cvrf/v3.0/cvrf/2026-Apr	T1
CVE-2026-32211 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-32211	T1
CVE-2026-32211 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-32211	T3
CVE-2026-32211 | Mondoo Vulnerability Intelligence	https://mondoo.com/vulnerability-intelligence/vulnerability/CVE-202...	T3

Generated 2026-04-30 14:08 UTC by TJS Security Command Center