INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-10 13:12 UTC

Three Ollama Vulnerabilities Expose AI Infrastructure to Memory Theft and Persistent Code Execution

CVE VULNERABILITY | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0152
Type	CVE Vulnerability
CVE ID	CVE-2026-7482, CVE-2026-42248, CVE-2026-42249
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.0011 (28th percentile)
Affected Products	Ollama (open-source LLM framework), versions prior to 0.17.1 (CVE-2026-7482); Ollama for Windows versions 0.12.10 through 0.22.0 (CVE-2026-42248, CVE-2026-42249)
Published	2026-05-10T08:41:00
Discovery Source	Rss

Executive Summary

Three vulnerabilities in Ollama, a widely deployed open-source AI framework, expose organizations running local large language models to memory theft and persistent code execution. The most severe flaw allows unauthenticated remote attackers to read heap memory from Ollama servers, potentially leaking API keys, credentials, and conversation data from AI workloads. Two Windows-specific vulnerabilities remain unpatched and can be chained to establish persistent code execution, threatening any organization that has deployed Ollama on Windows infrastructure.

Technical Analysis

Three CVEs affect Ollama, an open-source LLM serving framework with an estimated 300,000+ internet-exposed instances.

CVE-2026-7482 (CWE-125: Out-of-Bounds Read, CVSS 9.1, rated High): Affects Ollama versions prior to 0.17.1. An unauthenticated remote attacker can trigger an out-of-bounds heap read, leaking memory contents that may include API keys, session tokens, model conversation data, and other credentials resident in the Ollama process heap. No authentication is required to exploit this flaw.

CVE-2026-42248 and CVE-2026-42249 (Windows-specific, Ollama for Windows versions 0.12.10 through 0.22.0, no patch available as of disclosure): These two flaws target Ollama's update mechanism. CWE mappings indicate path traversal (CWE-22), improper verification of cryptographic signatures (CWE-347), and missing authentication (CWE-306) as contributing weaknesses. A chained attack against the update mechanism enables persistent code execution at user privilege, consistent with MITRE techniques T1574.010 (Services File Permissions Weakness), T1547.001 (Registry Run Keys / Startup Folder), and T1543 (Create or Modify System Process).

ATT&CK coverage across all three CVEs: T1190 (Exploit Public-Facing Application), T1552 / T1552.001 (Unsecured Credentials / Credentials in Files), T1041 (Exfiltration Over C2 Channel), T1195 (Supply Chain Compromise), T1574 (Hijack Execution Flow).

Note on CVSS: Per-CVE CVSS scores for CVE-2026-42248 and CVE-2026-42249 are not independently confirmed in source data; the cluster is assessed as Critical (9.5) based on chaining potential and unpatched status, with CVE-2026-7482 individually rated as High (9.1).

Patch status: CVE-2026-7482 is patched in Ollama 0.17.1. CVE-2026-42248 and CVE-2026-42249 have no patch available for Windows as of disclosure date.

Action Checklist

1. Step 1: Containment. Identify all Ollama deployments in your environment. Block unauthenticated external access to Ollama's default port (11434/tcp) at the network perimeter immediately. For Windows deployments running versions 0.12.10 through 0.22.0, isolate the host from the internet until a patch is available. Audit whether any Ollama instance is internet-facing without authentication or a reverse proxy enforcing access control.
2. Step 2: Detection. Query asset inventory and EDR for Ollama process execution (ollama.exe on Windows, ollama binary on Linux/macOS). Review firewall and proxy logs for inbound connections to port 11434 from external IPs. On Windows hosts, review Startup folder entries (%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup), registry Run keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run), and scheduled tasks for unexpected Ollama-related entries consistent with T1547.001 and T1543. Review process creation logs for ollama.exe spawning unexpected child processes. Check for anomalous outbound connections from the Ollama process host, consistent with T1041.
3. Step 3: Eradication. For CVE-2026-7482: upgrade all Ollama instances to version 0.17.1 or later immediately via the official Ollama release channel (https://github.com/ollama/ollama/releases). For CVE-2026-42248 and CVE-2026-42249: no patch is available. Mitigate by restricting Ollama on Windows to localhost-only binding, disabling the auto-update mechanism until a fix is released, and preventing internet access from the Ollama host. Remove any persistence mechanisms (registry keys, startup entries) added without authorization.
4. Step 4: Recovery. After patching CVE-2026-7482, confirm the running Ollama version with 'ollama --version' or equivalent. Rotate all API keys, service credentials, and tokens that were accessible to the Ollama process or stored in its working environment; treat heap-resident secrets as compromised. Monitor Ollama process behavior and outbound network activity for 72 hours post-remediation. For Windows systems where CVE-2026-42248/42249 mitigation was applied, verify the update mechanism remains disabled and no new persistence entries have appeared.

5. Step 5: Post-Incident. This incident exposes three control gaps: (1) AI/ML infrastructure is frequently deployed without the same access controls applied to production web services, enforce authentication on all LLM API endpoints as a baseline standard; (2) Windows software update mechanisms are an under-reviewed attack surface, add update mechanism integrity to application security assessments; (3) Secret management hygiene: API keys and credentials should not be accessible to LLM serving processes unless explicitly required, review secrets management for AI workloads and apply least-privilege principles. Add Ollama and similar LLM serving frameworks to your vulnerability management inventory if not already tracked.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and legal/privacy counsel immediately if forensic review of Ollama access logs or network flows reveals external IP connections to port 11434 prior to containment, as successful exploitation of CVE-2026-7482 may constitute a reportable data breach if heap-resident data included PII, PHI, or regulated credentials accessible to the LLM serving process.
Recovery Notes	After patching CVE-2026-7482 to Ollama 0.17.1+, verify the running version on every host in the inventory and confirm no Ollama instance remains bound to 0.0.0.0 on port 11434 without an authenticating reverse proxy in front of it. For Windows hosts under CVE-2026-42248/42249 mitigation, conduct daily Autoruns diffs for a minimum of 14 days given the unpatched persistence vectors, and treat any new HKCU Run key or Startup folder entry on those hosts as a presumptive indicator of compromise until proven otherwise. Treat all API keys and tokens that were in scope of the Ollama process environment as fully compromised regardless of whether exploitation is confirmed — the heap memory disclosure nature of CVE-2026-7482 makes silent theft undetectable after the fact.
Forensic Artifacts	Ollama application logs at ~/.ollama/logs/ (Linux/macOS) or %LOCALAPPDATA%\Ollama\logs\ (Windows): review for anomalous API request volumes, malformed requests, or repeated calls to /api/generate and /api/chat from external source IPs that may indicate CVE-2026-7482 heap probing activity. | Windows registry hive export of HKCU\Software\Microsoft\Windows\CurrentVersion\Run and HKLM\Software\Microsoft\Windows\CurrentVersion\Run: any Ollama-related or unnamed entries added during the CVE-2026-42248/42249 exploitation window represent persistence artifacts that must be preserved before eradication. | Sysmon Event ID 1 (Process Create) records for ollama.exe with full CommandLine and ParentCommandLine fields: unexpected child processes (cmd.exe, powershell.exe, wscript.exe) spawned by ollama.exe are direct indicators of code execution via CVE-2026-42248 or CVE-2026-42249 on Windows. | Network flow or firewall logs showing inbound TCP connections to port 11434 from non-RFC1918 source addresses: connection frequency, payload size distribution, and session duration anomalies are the primary forensic indicators of CVE-2026-7482 heap memory read exploitation, since the vulnerability is unauthenticated and leaves no application-level authentication trail. | Contents of %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup directory with file metadata (creation timestamp, owner, SHA-256 hash): files placed here by CVE-2026-42248/42249 exploitation for persistence will typically show creation timestamps correlated with Ollama process activity and owner accounts inconsistent with the legitimate software installer.

Per-Action IR Details

Step 1: Containment — Identify all Ollama deployments in your environment. Block unauthenticated external access to Ollama's default port (11434/tcp) at the network perimeter immediately. For Windows deployments running versions 0.12.10 through 0.22.0, isolate the host from the internet until a patch is available. Audit whether any Ollama instance is internet-facing without authentication or a reverse proxy enforcing access control.

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 12.2 — Establish and Maintain a Secure Network Architecture (restrict Ollama port 11434/tcp to internal segments only)

Compensating: Two-person team without enterprise tooling: run 'netstat -anp | grep 11434' (Linux) or 'netstat -ano | findstr 11434' (Windows) on suspected hosts to confirm active listeners. Use Windows Firewall with Advanced Security (wf.msc) or iptables/nftables to block inbound 11434/tcp from non-RFC1918 addresses immediately. For asset discovery across subnets, run 'nmap -p 11434 --open 192.168.0.0/16' (adjust CIDR) to enumerate any internet-reachable Ollama instances before perimeter rules are confirmed. On Windows hosts in the 0.12.10–0.22.0 range, pull the network cable or disable the NIC via Device Manager as an emergency measure if firewall rule propagation will be delayed.

Evidence: Before blocking port 11434, capture a full packet capture of current inbound connections to that port using 'tcpdump -i eth0 -w ollama_pre_block_$(date +%s).pcap port 11434' (Linux) or Wireshark with capture filter 'tcp port 11434' (Windows). Preserve netstat output showing all established connections to 11434 to identify source IPs that may have already exploited CVE-2026-7482. Snapshot 'ss -tnp sport = :11434' (Linux) or 'Get-NetTCPConnection -LocalPort 11434' (PowerShell) to record PIDs and parent process relationships before containment disrupts the running state.

Step 2: Detection — Query asset inventory and EDR for Ollama process execution (ollama.exe on Windows, ollama binary on Linux/macOS). Review firewall and proxy logs for inbound connections to port 11434 from external IPs. On Windows hosts, review Startup folder entries (%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup), registry Run keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run), and scheduled tasks for unexpected Ollama-related entries consistent with T1547.001 and T1543. Review process creation logs for ollama.exe spawning unexpected child processes. Check for anomalous outbound connections from the Ollama process host, consistent with T1041.

NIST Phase: Detection Analysis

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

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

Compensating: Deploy Sysmon with the SwiftOnSecurity config (or Olaf Hartong's modular config) to capture Event ID 1 (Process Create) for ollama.exe and Event ID 3 (Network Connection) for outbound connections from the Ollama process. Query collected Sysmon logs with: 'Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational" | Where-Object {$_.Message -like "*ollama*"}'. For persistence hunting without EDR, run 'reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Run' and 'schtasks /query /fo LIST /v | findstr /i "ollama"' manually on each Windows host. Use osquery with 'SELECT * FROM startup_items WHERE name LIKE "%ollama%";' and 'SELECT * FROM scheduled_tasks WHERE action LIKE "%ollama%";' for cross-host sweeps. For heap memory exfiltration detection (CVE-2026-7482), review web server or reverse proxy access logs for repeated GET/POST requests to Ollama API endpoints (/api/generate, /api/chat) originating from unexpected external IPs.

Evidence: Collect Windows Security Event Log Event ID 4688 (Process Creation with command-line logging enabled) filtering on ParentProcessName containing 'ollama.exe' to identify any child processes spawned post-exploitation. Export Sysmon Event ID 1 records for ollama.exe with full command-line arguments to detect unusual flags or injected parameters. For CVE-2026-7482 heap memory theft, preserve Ollama application logs (default path: ~/.ollama/logs/ on Linux, %LOCALAPPDATA%\Ollama\logs\ on Windows) which may record anomalous API request volumes or malformed request patterns from attacker IPs. Capture the contents of %APPDATA%\Microsoft\Windows\Start

Menu\Programs\Startup and export 'HKCU\Software\Microsoft\Windows\CurrentVersion\Run' registry hive snapshot before any remediation. For T1041 exfiltration, extract Zeek/firewall flow logs showing outbound connections from the Ollama host process PID to non-standard destinations.

Step 3: Eradication — For CVE-2026-7482: upgrade all Ollama instances to version 0.17.1 or later immediately via the official Ollama release channel (https://github.com/ollama/ollama/releases). For CVE-2026-42248 and CVE-2026-42249: no patch is available. Mitigate by restricting Ollama on Windows to localhost-only binding, disabling the auto-update mechanism until a fix is released, and preventing internet access from the Ollama host. Remove any persistence mechanisms (registry keys, startup entries) added without authorization.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST IR-4 (Incident Handling), NIST CM-7 (Least Functionality), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: For CVE-2026-7482 patching verification without a software management platform: after upgrading, run 'ollama --version' and confirm output shows 0.17.1 or higher; script this check across hosts with 'for host in $(cat hosts.txt); do ssh $host "ollama --version"; done'. For CVE-2026-42248/42249 localhost-only binding on Windows without enterprise config management: set the OLLAMA_HOST environment variable to '127.0.0.1' in the Windows system environment (System Properties → Environment Variables) and restart the Ollama service. Disable the Ollama auto-update mechanism by blocking outbound DNS/HTTP for update endpoints using Windows Firewall: 'netsh advfirewall firewall add rule name="Block Ollama Update" dir=out action=block program="%LOCALAPPDATA%\Programs\Ollama\ollama.exe" remoteip=any'. For unauthorized persistence removal: use Autoruns (Sysinternals, free) to enumerate and remove all Ollama-related startup entries, Run keys, and scheduled tasks — hash-verify legitimate entries against the known-good Ollama installer hash from the GitHub releases page before deletion.

Evidence: Before removing persistence artifacts for CVE-2026-42248/42249, image the full registry hive 'HKCU\Software\Microsoft\Windows\CurrentVersion\Run' and the Startup folder contents using 'reg export HKCU\Software\Microsoft\Windows\CurrentVersion\Run ollama_runkey_evidence.reg'. Collect file metadata (creation time, last modified, owner) for any unauthorized entries in %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup using 'Get-ChildItem -Path "$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup" | Select-Object Name, CreationTime, LastWriteTime, Owner | Export-Csv startup_evidence.csv'. Capture the full list of scheduled tasks before deletion: 'schtasks /query /xml > scheduled_tasks_evidence.xml'. Hash the suspected malicious files with 'Get-FileHash -Algorithm SHA256 ' for threat intelligence comparison before deletion.

Step 4: Recovery — After patching CVE-2026-7482, confirm the running Ollama version with 'ollama --version' or equivalent. Rotate all API keys, service credentials, and tokens that were accessible to the Ollama process or stored in its working environment — treat heap-resident secrets as compromised. Monitor Ollama process behavior and outbound network activity for 72 hours post-remediation. For Windows systems where CVE-2026-42248/42249 mitigation was applied, verify the update mechanism remains disabled and no new persistence entries have appeared.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST IA-5 (Authenticator Management), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 5.2 (Use Unique Passwords)

Compensating: Credential rotation without a secrets management platform: enumerate all environment variables accessible to the Ollama process at the time of exposure using 'strings /proc/$(pgrep ollama)/environ' (Linux) or 'Get-Process ollama | ForEach-Object { [System.Diagnostics.Process]::GetProcessById($_.Id).StartInfo.EnvironmentVariables }' (Windows, if process is still running in a test environment) to identify exactly which secrets were heap-resident. Prioritize rotation of any API keys

found in ollama environment config files (~/.ollama/config or %APPDATA%\ollama\config). For 72-hour behavioral monitoring without EDR: configure a Sysmon Event ID 3 alert rule for new outbound network connections from ollama.exe to non-localhost destinations, reviewed via 'Get-WinEvent -LogName Microsoft-Windows-Sysmon/Operational' on a scheduled task every 4 hours. Verify update mechanism remains disabled by checking Windows Firewall rules daily: 'netsh advfirewall firewall show rule name="Block Ollama Update"'.

Evidence: Document the patch state with 'ollama --version' output timestamped and signed (hash the terminal output file). Preserve a list of all rotated credentials with rotation timestamps (not the credential values) for the incident record per NIST IR-5 (Incident Monitoring) requirements. Capture a baseline Sysmon Event ID 3 network connection log for the Ollama process immediately post-patch to establish the expected outbound connection profile for comparison during the 72-hour monitoring window. For Windows CVE-2026-42248/42249 hosts, run Autoruns at 24-hour and 72-hour marks and diff the output against the post-eradication baseline to detect re-establishment of persistence.

Step 5: Post-Incident — This incident exposes three control gaps: (1) AI/ML infrastructure is frequently deployed without the same access controls applied to production web services — enforce authentication on all LLM API endpoints as a baseline standard; (2) Windows software update mechanisms are an under-reviewed attack surface — add update mechanism integrity to application security assessments; (3) secret management hygiene: API keys and credentials should not be accessible to LLM serving processes unless explicitly required — review secrets management for AI workloads and apply least-privilege principles. Add Ollama and similar LLM serving frameworks to your vulnerability management inventory if not already tracked.

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 SI-5 (Security Alerts, Advisories, and Directives), NIST RA-3 (Risk Assessment), NIST SA-22 (Unsupported System Components), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: For teams without a formal asset management platform: add Ollama and all LLM serving processes (llama.cpp, LM Studio, LocalAI) to a tracked software inventory in a simple CSV or spreadsheet, monitored against the GitHub releases RSS feed (https://github.com/ollama/ollama/releases.atom) for new CVEs — subscribe this feed in any RSS reader or pipe it to a Slack webhook for free alerting. For secrets management gap remediation without a vault: audit Ollama config files (~/.ollama/ and %APPDATA%\ollama\) for hardcoded API keys using a one-time grep: 'grep -r "sk-\|api_key\|token" ~/.ollama/' and migrate identified secrets to OS-level credential stores (Windows Credential Manager, Linux GNOME Keyring) or environment variables scoped only to the Ollama service account, not the interactive user. Document update mechanism integrity checks as a line item in the next application security assessment template.

Evidence: Produce a lessons-learned document capturing: (1) the count and locations of all Ollama instances discovered during Step 1 versus what was in the asset inventory — this delta quantifies the shadow AI infrastructure gap; (2) the list of credential types confirmed or suspected as heap-resident during CVE-2026-7482 exposure (API keys, session tokens, model inference credentials); (3) the time-to-detect and time-to-contain metrics against the organization's IR SLA targets per NIST 800-61r3 §4. Retain all forensic evidence collected in Steps 1–4 per NIST AU-11 (Audit Record Retention) for the organization's defined retention period, as heap memory theft of credentials may trigger delayed downstream account compromise incidents requiring this evidence chain.

Detection Guidance

Primary indicators to investigate:

1. Network: Scan for assets with port 11434/tcp open to the internet. Alert on inbound connections to that port from non-approved external IPs. Correlate with any Ollama process outbound connections that are anomalous in volume or destination.

2. Process telemetry (Windows): Look for ollama.exe spawning cmd.exe, powershell.exe, or other shells as child processes. Look for ollama.exe writing to user startup directories or modifying Run registry keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run, HKLM equivalent if privilege allows).

3. Persistence artifacts (Windows, CVE-2026-42248/42249 specific): Check %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup and Task Scheduler for Ollama-related entries created outside of a known installation event. Cross-reference creation timestamps against Ollama installation date.

4. Memory/credential exposure (CVE-2026-7482): If any Ollama instance was internet-accessible before patching to 0.17.1, treat all API keys and credentials loaded into that process as potentially compromised. There is no reliable post-hoc log-based detection for heap read exploitation; assume exposure if the instance was reachable.

5. SIEM query starting point: Filter for process creation events where ParentImage contains 'ollama' and ChildImage is a shell interpreter. Filter for file writes by ollama.exe to startup paths. Alert on outbound network connections from ollama.exe to non-local destinations on unexpected ports.

ATT&CK techniques to map detections against: T1190, T1552.001, T1547.001, T1543, T1041, T1574.010.

Framework Mappings

MITRE-ATTACK

- **T1574.010** — Services File Permissions Weakness
- **T1195** — Supply Chain Compromise
- **T1190** — Exploit Public-Facing Application
- **T1552** — Unsecured Credentials
- **T1552.001** — Credentials In Files
- **T1574** — Hijack Execution Flow
- **T1543** — Create or Modify System Process
- **T1041** — Exfiltration Over C2 Channel
- **T1547.001** — Registry Run Keys / Startup Folder

NIST-800-53R5

- **SA-9** — External System Services
- **SR-2** — Supply Chain Risk Management Plan
- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **CA-7** — Continuous Monitoring
- **SI-4** — System Monitoring

- **SI-16** — Memory Protection
- **IA-2** — Identification and Authentication (Organizational Users)
- **AC-3** — Access Enforcement
- **SI-10** — Information Input Validation
- **SC-13** — Cryptographic Protection

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.3** — Require MFA for Externally-Exposed Applications
- **16.12** — Implement Code-Level Security Checks
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures
- **A01:2021** — Broken Access Control

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information
- **A.8.24** — Use of cryptography
- **A.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1574.010	Services File Permissions Weakness	Persistence
T1195	Supply Chain Compromise	Initial-Access
T1190	Exploit Public-Facing Application	Initial-Access
T1552	Unsecured Credentials	Credential-Access
T1552.001	Credentials In Files	Credential-Access
T1574	Hijack Execution Flow	Persistence
T1543	Create or Modify System Process	Persistence

Technique ID	Technique Name	Tactic
T1041	Exfiltration Over C2 Channel	Exfiltration
T1547.001	Registry Run Keys / Startup Folder	Persistence

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/05/ollama-out-of-bounds-read-vulnera...	T3
CVE-2026-42249 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-42249	T1
CVE-2026-7482 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-7482	T3
CVE-2026-42248 - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-42248	T1
[PDF] Security Bulletin 06 May 2026	https://isomer-user-content.by.gov.sg/36/463c774c-c6c3-46fb-8d25-68...	T1
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-7482, CVE-2026-42248, CVE...	T1

Generated 2026-05-10 13:12 UTC by TJS Security Command Center