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

CVE-2026-7022: A security vulnerability has been detected in SmythOS sre up to 0.0.15. Affected is the function Age...

CVE VULNERABILITY | HIGH | CVSS 7.3

SCC Item ID	SCC-CVE-2026-0093
Type	CVE Vulnerability
CVE ID	CVE-2026-7022
Severity	HIGH
CVSS Base Score	7.3
EPSS Score	0.0006 (19th percentile)
Affected Products	SmythOS sre <= 0.0.15
Published	2026-04-26T06:16:02.210
Discovery Source	Nvd

Executive Summary

A publicly disclosed authentication bypass vulnerability (CVE-2026-7022) affects SmythOS sre versions 0.0.15 and earlier, an AI agent runtime platform. Remote attackers with no credentials can manipulate HTTP headers to bypass authentication controls entirely. Organizations running exposed SmythOS sre instances face unauthorized access to AI agent infrastructure until patched or mitigated.

Technical Analysis

CVE-2026-7022 is a CWE-287 (Improper Authentication) vulnerability in SmythOS sre <= 0.0.15. The flaw resides in the AgentRuntime class (packages/core/src/subsystems/AgentManager/AgentRuntime.class.ts), specifically in the HTTP Header Handler component. Manipulation of the X-DEBUG-RUN and X-DEBUG-INJ HTTP headers allows unauthenticated remote attackers to bypass authentication controls without valid credentials. CVSS base score: 7.3 (High); CVSS vector pending NVD publication. EPSS score: 0.00061 (18.97th percentile, low current exploitation probability). The exploit has been publicly disclosed. The vendor was contacted prior to disclosure and did not respond (per GHSA-rmc7-qc5q-h96j). No official vendor patch has been published at this time. MITRE ATT&CK mappings: T1190 (Exploit Public-Facing Application), T1556 (Modify Authentication Process). No CISA KEV listing as of configuration date.

Action Checklist

1. Step 1: Containment, Immediately restrict network access to SmythOS sre instances (versions <= 0.0.15). Place internet-facing deployments behind a WAF or firewall rule blocking requests containing X-DEBUG-RUN and X-DEBUG-INJ headers. If public exposure cannot be immediately removed, take the service offline until remediation is complete.
2. Step 2: Detection, Review HTTP access logs for requests containing the headers X-DEBUG-RUN or X-DEBUG-INJ targeting SmythOS sre endpoints. Query your SIEM or log aggregator for these header values across all ingress points. Flag any 200/302 responses to such requests as potential exploitation attempts. Check the GitHub advisory (GHSA-rmc7-qc5q-h96j) for any published proof-of-concept request signatures.
3. Step 3: Eradication, Upgrade SmythOS sre to a version above 0.0.15 when a patched release becomes available from the vendor. In the absence of a vendor patch, implement server-side or WAF rules to strip or reject requests containing X-DEBUG-RUN and X-DEBUG-INJ headers before they reach the AgentRuntime component. Remove or disable debug header handling in the AgentRuntime.class.ts if source access is available.
4. Step 4: Recovery, After applying controls, verify authentication enforcement by testing requests with the affected headers and confirming they are rejected with 401/403 responses. Review audit logs for any unauthorized access that may have occurred prior to mitigation. Restore service only after confirming header-based bypass is blocked.
5. Step 5: Post-Incident, Document the exposure window and scope of any unauthorized access. Review the authentication architecture of other AI agent runtime components for similar debug-mode header patterns. Configure alerts or calendar reminders to check for vendor patches and public advisory updates weekly until a patch is released or the vulnerability is formally retracted.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to senior IR leadership and legal/compliance if HTTP access logs confirm any 200/302 responses to X-DEBUG-RUN or X-DEBUG-INJ header requests during the exposure window, as confirmed exploitation of this authentication bypass on an AI agent runtime platform may constitute unauthorized access to automated business process infrastructure and could trigger breach notification obligations under applicable data protection regulations (GDPR, CCPA, HIPAA) depending on data processed by the AI agents.
Recovery Notes	After containment and eradication, monitor SmythOS sre HTTP access logs and AgentRuntime application logs continuously for at least 72 hours post-restoration for any recurrence of X-DEBUG-RUN or X-DEBUG-INJ header requests, which may indicate an attacker retesting for the bypass or probing for new debug header variants in the same codebase. Verify that all AI agent tasks executed during the exposure window were legitimate by auditing AgentRuntime task execution logs against authorized user activity; any orphaned or unattributed agent task executions should be treated as potentially attacker-initiated and investigated for data exfiltration or lateral movement within the AI agent infrastructure. Do not restore full public exposure of the SmythOS sre instance until a vendor-issued patch above version 0.0.15 is applied and independently verified.

Forensic Artifacts	Web server access logs (e.g., /var/log/nginx/access.log or /var/log/apache2/access.log): Primary evidence source — filter for requests containing 'X-DEBUG-RUN' or 'X-DEBUG-INJ' header values with HTTP 200/302 response codes, which directly confirm CVE-2026-7022 exploitation attempts and successes against the AgentRuntime endpoints. | SmythOS sre AgentRuntime application log: Records debug-mode activation events and agent task executions triggered under the bypassed authentication context — critical for determining whether unauthenticated sessions resulted in actual AI agent task execution, not merely authentication bypass. | OS process audit log for the sre service user: Query for child processes spawned by the SmythOS sre process user during the exposure window (Linux: `ausearch -c sre` or `journalctl _UID=`) — unauthorized agent task execution may have spawned subprocesses for data retrieval, API calls, or code execution depending on the agent's configured capabilities. | Network flow records (NetFlow/IPFIX or connection logs) for the sre service port: Sustained or high-volume sessions from external IPs to the sre port following a successful header bypass response indicate active exploitation beyond initial access, such as attacker-directed agent task chains. | Preserved copy of AgentRuntime.class.ts (SHA-256 hashed) from the vulnerable deployment: Documents the exact vulnerable code path for the debug header bypass in the AgentRuntime component, serves as evidence of the root cause, and provides the baseline for confirming the patch or code remediation actually removed the vulnerable logic.

Per-Action IR Details

Step 1: Containment — Immediately restrict network access to SmythOS sre instances (versions <= 0.0.15). Place internet-facing deployments behind a WAF or firewall rule blocking requests containing X-DEBUG-RUN and X-DEBUG-INJ headers. If public exposure cannot be immediately removed, take the service offline until remediation is complete.

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 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Using iptables or nftables, immediately drop inbound HTTP/HTTPS traffic to the SmythOS sre port from external IPs: `iptables -I INPUT -p tcp --dport -j DROP`. If WAF is unavailable, deploy ModSecurity (free, Apache/Nginx module) with a custom rule: `SecRule REQUEST_HEADERS:X-DEBUG-RUN "@rx .+" "id:100001,phase:1,deny,status:403,msg:'CVE-2026-7022 header blocked'"` and a parallel rule for X-DEBUG-INJ. A 2-person team can deploy this in under 30 minutes on a Linux host.

Evidence: Before isolating the instance, capture a full snapshot of active HTTP connections to the SmythOS sre service: `ss -tnp | grep ` to record active sessions. Export the current web server access log (e.g., /var/log/nginx/access.log or /var/log/apache2/access.log) and any reverse proxy logs. If the service runs in a container, capture `docker logs ` before stopping it. Document the external IP addresses and timestamps of any recent connections for later threat actor attribution.

Step 2: Detection — Review HTTP access logs for requests containing the headers X-DEBUG-RUN or X-DEBUG-INJ targeting SmythOS sre endpoints. Query your SIEM or log aggregator for these header values across all ingress points. Flag any 200/302 responses to such requests as potential exploitation attempts. Check the GitHub advisory (GHSA-rmc7-qc5q-h96j) for any published proof-of-concept request signatures.

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), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, run this grep against the web server access log to surface all requests carrying either debug header: `grep -E 'X-DEBUG-RUN|X-DEBUG-INJ' /var/log/nginx/access.log | awk '{print $1, $7, $9}'` to extract source IP, requested URI, and HTTP response code. Pipe through `| grep -E ' 200 | 302 '` to isolate successful bypass responses. For ongoing detection, deploy a Sigma rule targeting HTTP access logs filtering on these two header strings with a 2xx/3xx response threshold, convertible to a cron-based grep alert for teams without a SIEM. Cross-reference source IPs against abuse databases using `curl https://api.abuseipdb.com/api/v2/check?ipAddress=` with a free AbuseIPDB API key.

Evidence: The primary forensic artifact for exploitation of CVE-2026-7022 is the HTTP request log entry showing X-DEBUG-RUN or X-DEBUG-INJ header values paired with a 200 or 302 response code directed at SmythOS AgentRuntime endpoints (likely paths such as /agent, /run, or /execute based on the AgentRuntime.class.ts component). Also capture: any application-level logs generated by the AgentRuntime component itself (check the SmythOS sre application log directory for debug-mode activation events), network flow records (NetFlow/IPFIX) showing sustained sessions from external IPs to the sre port after the header-based bypass, and any spawned child processes if the AI agent runtime executed tasks under the unauthenticated session (check OS process audit logs for processes spawned by the sre service user).

Step 3: Eradication — Upgrade SmythOS sre to a version above 0.0.15 when a patched release becomes available from the vendor. In the absence of a vendor patch, implement server-side or WAF rules to strip or reject requests containing X-DEBUG-RUN and X-DEBUG-INJ headers before they reach the AgentRuntime component. Remove or disable debug header handling in the AgentRuntime.class.ts if source access is available.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality), 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: If vendor patch is not yet available and source access exists, locate the debug header parsing block in AgentRuntime.class.ts and comment out or remove the conditional logic that branches execution based on X-DEBUG-RUN or X-DEBUG-INJ header presence. After modification, restart the service and validate that requests with those headers return 401/403 (see Step 4). If source modification is not possible, configure Nginx to strip these headers before proxying to the sre backend: `proxy_set_header X-DEBUG-RUN ""; proxy_set_header X-DEBUG-INJ "";` — this prevents the headers from reaching the AgentRuntime component entirely. Document the code change or proxy config as a formal compensating control with a review date tied to vendor patch availability.

Evidence: Before applying the patch or code change, preserve a copy of the unmodified AgentRuntime.class.ts (or its compiled equivalent) as forensic evidence of the vulnerable code path. Capture a file hash (SHA-256) of the vulnerable version: `sha256sum AgentRuntime.class.ts`. If the environment was potentially compromised during the exposure window, also collect a memory dump of the running sre process (`gcore ` on Linux) before terminating it, as unauthorized AI agent task execution may have left in-memory artifacts (spawned agent configurations, injected task parameters) that are lost on process termination.

Step 4: Recovery — After applying controls, verify authentication enforcement by testing requests with the affected headers and confirming they are rejected with 401/403 responses. Review audit logs for any unauthorized access that may have occurred prior to mitigation. Restore service only after confirming header-based bypass is blocked.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-6 (Security and Privacy Function Verification), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Perform functional verification using curl: `curl -v -H 'X-DEBUG-RUN: true' -H 'X-DEBUG-INJ: true' https://:/agent` — a patched or mitigated instance must return HTTP 401 or 403; any 200/302 indicates the bypass is still active. Run this test from an external IP (or simulate via a different network segment) to confirm the header

stripping or rejection occurs at the perimeter, not just internally. Log the test request and response as documented evidence of remediation verification per NIST SI-2 requirements.

Evidence: Pull the full HTTP access log covering the exposure window (from the earliest instance of SmythOS sre 0.0.15 deployment to the time containment controls were applied) and archive it to write-once storage or a hash-verified archive (`tar czf - /var/log/nginx/access.log | tee access_log_CVE-2026-7022.tar.gz | sha256sum`). Cross-reference any 200/302 responses to X-DEBUG-RUN/X-DEBUG-INJ requests against the AgentRuntime application log to determine whether unauthorized agent tasks were actually executed — this distinction separates a probe/scan event from a confirmed exploitation event and drives breach notification decisions.

Step 5: Post-Incident — Document the exposure window and scope of any unauthorized access. Review the authentication architecture of other AI agent runtime components for similar debug-mode header patterns. Establish a process for monitoring vendor response to disclosed vulnerabilities, especially for components that did not respond to prior disclosure contact.

NIST Phase: Post Incident

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

Controls: NIST IR-4 (Incident Handling), NIST IR-5 (Incident Monitoring), NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For the architectural review of other AI agent runtime components, audit all HTTP middleware and routing code for similar patterns: grep the codebase for header-gated debug or bypass logic: `grep -rn 'X-DEBUG\|debug.*header\|bypass.*auth\|req.headers' ./src/`. For vendor monitoring, configure a GitHub release watcher on the SmythOS sre repository and a CVE feed subscription filtered to 'SmythOS' via CISA's Known Exploited Vulnerabilities catalog RSS feed or OSV.dev API query (`https://api.osv.dev/v1/query` with package ecosystem filter). A 2-person team can automate this with a weekly cron job posting results to a shared Slack channel or email alias.

Evidence: Compile the final incident timeline artifact: the exposure window (first deployment date of sre <= 0.0.15 through confirmed mitigation timestamp), the complete set of source IPs that sent X-DEBUG-RUN or X-DEBUG-INJ headers with successful responses, and any AgentRuntime application logs showing what actions were taken under unauthenticated sessions. This package supports both internal lessons-learned and any regulatory breach notification assessment. Retain all raw logs per NIST AU-11 (Audit Record Retention) requirements for your applicable retention period.

Detection Guidance

Search HTTP access logs and WAF logs for inbound requests containing the headers X-DEBUG-RUN or X-DEBUG-INJ directed at SmythOS sre endpoints. A SIEM query pattern (adapt to your log format): filter on http.request.headers matching 'X-DEBUG-RUN' OR 'X-DEBUG-INJ' with a destination matching known SmythOS sre service addresses. Flag any such requests that received a successful response (HTTP 200, 302, or 201). Treat any confirmed match as a high-priority event requiring manual review given the public disclosure. No public IOC hashes or IP indicators are available at this time from the listed sources.

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1556** — Modify Authentication Process

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
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **SI-4** — System Monitoring
- **IA-8** — Identification and Authentication (Non-Organizational Users)

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets
- **CC9.2** — Manages risks associated with vendors and business partners

HIPAA-SECURITY

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

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1190	Exploit Public-Facing Application	Initial-Access
T1556	Modify Authentication Process	Credential-Access

Sources

Source	URL	Tier
nvd	https://nvd.nist.gov/vuln/detail/CVE-2026-7022	T1
CVE-2026-7022 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-7022	T3
CVE Alert: CVE-2026-7022 - SmythOS - sre - RedPacket Security	https://www.redpacketsecurity.com/cve-alert-cve-2026-7022-smythos-sre/	T3
A security vulnerability has been detected in SmythOS sre... - GitHub	https://github.com/advisories/GHSA-rmc7-qc5q-h96j	T3
New CVE Alert: CVE-2026-7022 Severity: 7.3 Risk Level: High ...	https://x.com/CVEarity/status/2048382236618379739	T3

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