INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-17 06:49 UTC

Apache ActiveMQ Jolokia RCE: 13-Year-Old Attack Surface Now Under Active Exploitation with Federal Patch Deadline

CVE VULNERABILITY | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0046
Type	CVE Vulnerability
CVE ID	CVE-2026-34197, CVE-2024-32114, CVE-2023-46604
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.0622 (91th percentile)
Affected Products	Apache ActiveMQ Classic (org.apache.activemq:activemq-broker and activemq-all) versions prior to 5.19.4 and 6.0.0-6.2.2
Published	2026-04-16T23:22:00
Discovery Source	Rss

Executive Summary

A critical remote code execution vulnerability (CVE-2026-34197) in Apache ActiveMQ Classic is under active exploitation, confirmed by CISA's addition to the Known Exploited Vulnerabilities catalog with a mandatory federal remediation deadline of April 30, 2026. Attackers can abuse the Jolokia management API to execute arbitrary commands on affected messaging infrastructure. On versions 6.0.0-6.1.1, the vulnerability is unauthenticated and network-accessible, requiring no credentials; when chained with a missing-authentication flaw (CVE-2024-32114), exploitation is trivial. Organizations running ActiveMQ in enterprise messaging, data pipelines, or integration middleware face immediate risk of data exfiltration, lateral movement, and service disruption.

Technical Analysis

CVE-2026-34197 is a critical-severity RCE in Apache ActiveMQ Classic affecting activemq-broker and activemq-all packages. Affected versions: all releases prior to 5.19.4 and 6.0.0 through 6.2.1 (inclusive). The vulnerability abuses the Jolokia JMX-over-HTTP management API (CWE-94: improper code control; CWE-20: improper input validation) to execute arbitrary OS commands on the host running the broker. On versions 6.0.0-6.1.1, CVE-2024-32114 (CWE-306: missing authentication on the Jolokia endpoint) chains with

CVE-2026-34197 to produce unauthenticated RCE, attack complexity drops to trivial with no prior access required. The Jolokia API surface has been present for 13+ years and is broadly exposed in enterprise deployments. EPSS score: 0.062 (90.9th percentile), indicating elevated exploitation probability relative to the CVE population. CVSS base score of 9.5 is pending NVD official publication; qualitative rating of Critical is applied on the basis of confirmed active exploitation and CISA KEV inclusion. MITRE ATT&CK coverage: T1190 (Exploit Public-Facing Application), T1059 (Command and Scripting Interpreter), T1210 (Exploitation of Remote Services), T1041 (Exfiltration Over C2 Channel), T1071 (Application Layer Protocol), T1021 (Remote Services), T1078.001 (Default Accounts). A well-documented predecessor, CVE-2023-46604 (OpenWire RCE), demonstrates that ActiveMQ has been a recurring target for exploitation; organizations should treat this vulnerability class as part of a known attack surface. CWE-1392 (use of default credentials) is also listed, suggesting hardcoded or default credential abuse may be part of the attack chain. As of April 2026, no specific threat actor or campaign attribution has been published by CISA or security research community; exploitation is confirmed as widespread but unattributed. Fixed versions: 5.19.4+ and 6.2.2+.

Action Checklist

1. Step 1: Containment, Immediately identify all ActiveMQ Classic instances (activemq-broker, activemq-all) running versions below 5.19.4 or between 6.0.0 and 6.2.1. Block external access to the Jolokia HTTP endpoint (default port 8161, path /api/jolokia) at the network perimeter and host-based firewall. If blocking Jolokia is not operationally feasible, enforce strong authentication on the endpoint and restrict access to trusted internal management networks only. Disable unauthenticated access entirely. If running 6.0.0-6.1.1, treat the system as unauthenticated-RCE-exposed and isolate from production networks pending patch.
2. Step 2: Detection, Query SIEM and EDR for anomalous process execution spawned from the ActiveMQ broker JVM process (e.g., activemq.jar or java processes spawning shell interpreters: bash, sh, cmd.exe, powershell.exe). Review web/application server logs for unexpected HTTP POST or GET requests to /api/jolokia or /jolokia paths with exec or write operation types. Hunt for T1190 indicators: unusual outbound connections from ActiveMQ hosts, new scheduled tasks or cron entries, and lateral movement from broker hosts (T1021). Check for CVE-2023-46604 IOC patterns (OpenWire port 61616) as a baseline comparison for actor TTPs.
3. Step 3: Eradication, Upgrade Apache ActiveMQ Classic to version 5.19.4 (5.x branch) or 6.2.2 (6.x branch) per the Apache ActiveMQ project release page. If immediate patching is not possible: disable the Jolokia endpoint entirely in activemq.xml or jetty.xml by removing or commenting out the Jolokia servlet configuration; enforce authentication on the web console; rotate any default or weak credentials on the broker and management interfaces (address CWE-1392).
4. Step 4: Recovery, After patching, confirm the running version via the ActiveMQ web console or 'activemq --version'. Validate that the /api/jolokia endpoint returns 401/403 or is unreachable from untrusted networks. Re-enable monitoring and confirm no persistence mechanisms (cron, scheduled tasks, new user accounts, webshells in ActiveMQ's web directory) were established during the exploitation window. Review broker configuration for unauthorized changes.
5. Step 5: Post-Incident, Conduct a full inventory of JMX/Jolokia-enabled services across the environment; this attack surface predates CVE-2026-34197 by 13 years and may affect other products. Evaluate whether management interfaces (ActiveMQ web console, Jolokia, JMX RMI) are appropriately segmented from production and internet-facing networks. Add ActiveMQ version monitoring to your vulnerability management program. Map findings to NIST CSF ID.AM (asset management) and PR.AC (access

control) and document control gaps for the next GRC review cycle.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO, legal counsel, and breach notification review if forensic evidence confirms pre-patch exploitation (Jolokia exec calls in Jetty logs, java-spawned shells in process logs, or unauthorized files in $ACTIVEMQ_HOME/webapps/) on any ActiveMQ instance that processes, routes, or has network adjacency to systems handling PII, PHI, PCI-scoped data, or OT/ICS environments — CISA KEV status and CVSS 9.5 with active exploitation meet mandatory federal reporting thresholds under FISMA and may trigger state breach notification timelines.
Recovery Notes	After patching to 5.19.4 or 6.2.2, maintain elevated monitoring on ActiveMQ broker hosts for a minimum of 30 days: alert on any new child process spawned by the broker JVM, any new file written to $ACTIVEMQ_HOME/webapps/, and any outbound connection from the broker host to non-whitelisted IPs. Given that CVE-2023-46604 (OpenWire deserialization RCE, 2023) was actively exploited by ransomware groups including HelloKitty and TellYouThePass, threat actors with existing access to ActiveMQ broker infrastructure have demonstrated willingness to deploy ransomware payloads — validate that backup integrity is confirmed and that recovery point objectives are met before returning the broker to production traffic. Verify broker message queue integrity post-recovery, as an attacker with broker-level RCE may have tampered with queued messages or broker configuration to establish a persistent re-entry path via modified activemq.xml plugin definitions.
Forensic Artifacts	Jetty HTTP access log ($ACTIVEMQ_HOME/data/activemq.log or configured Jetty log path): Contains timestamped HTTP POST/GET requests to /api/jolokia/exec/ with MBean operation names — the primary forensic record of CVE-2026-34197 exploitation attempts and successes, including source IP, user-agent, and operation payload. \| ActiveMQ audit log ($ACTIVEMQ_HOME/data/audit.log): Records broker-level authentication events and management operations; on 6.0.0–6.1.1 systems, successful Jolokia operations with no authentication record confirm CVE-2024-32114 exploitation (unauthenticated access). | OS process creation logs (Sysmon Event ID 1 on Windows, Linux auditd EXECVE records or /var/log/auth.log): Documents any child processes spawned by the ActiveMQ broker JVM (java.exe/javaw.exe parentage) — bash, sh, cmd.exe, powershell.exe, or curl/wget children are high-confidence indicators of successful RCE via the Jolokia exec interface. | File system artifacts in $ACTIVEMQ_HOME/webapps/ and /tmp or %TEMP%: Webshells (*.jsp, *.jspx files with creation timestamps post-broker-start), downloaded payloads, or staged tools dropped via Jolokia-executed OS commands; compare file timestamps against the Jetty log exploitation window to establish the attack timeline. \| JVM heap dump ($ACTIVEMQ_HOME directory or /tmp/activemq_heap.hprof if captured pre-shutdown): May contain deserialized class objects, injected bytecode, or in-memory webshell artifacts loaded via the Jolokia exec interface that are not present on disk, critical for confirming fileless or memory-resident post-exploitation activity.

Per-Action IR Details

Step 1: Containment — Immediately identify all ActiveMQ Classic instances (activemq-broker, activemq-all) running versions below 5.19.4 or between 6.0.0 and 6.2.1. Block external access to the Jolokia HTTP endpoint (default port 8161, path /api/jolokia) at the network perimeter and host-based firewall. If running 6.0.0–6.1.1, treat the system as unauthenticated-RCE-exposed and isolate from production networks pending patch.

NIST Phase: Containment

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

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), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Run 'find / -name activemq.jar 2>/dev/null' and 'ps aux | grep activemq' on Linux hosts, or 'Get-Process | Where-Object {$_.Name -like "*java*"}' on Windows, to enumerate running broker instances. Identify version via 'activemq --version' or inspect the MANIFEST.MF inside activemq-all-*.jar ('unzip -p activemq-all-*.jar META-INF/MANIFEST.MF | grep Implementation-Version'). Block port 8161 immediately using iptables: 'iptables -I INPUT -p tcp --dport 8161 -j DROP' (Linux) or 'netsh advfirewall firewall add rule name="Block Jolokia 8161" protocol=TCP dir=in localport=8161 action=block' (Windows). For 6.0.0–6.1.1 hosts, also block OpenWire port 61616 at the perimeter pending patch, as unauthenticated RCE is trivially achievable without any credential barrier.

Evidence: Before blocking port 8161, capture a full netstat snapshot to document established connections to the Jolokia HTTP endpoint: 'ss -tnp sport = :8161' (Linux) or 'netstat -ano | findstr :8161' (Windows). Preserve the ActiveMQ installation directory listing (ls -la $ACTIVEMQ_HOME/webapps/ and $ACTIVEMQ_HOME/data/) to establish a clean-state baseline for later comparison. Capture running process tree ('ps auxf' on Linux, 'Get-CimInstance Win32_Process | Select ProcessId, ParentProcessId, Name, CommandLine' on Windows) to identify any child processes already spawned by the broker JVM — a java process with child bash/sh/cmd.exe is a strong indicator of pre-containment exploitation. Dump active network connections from the broker host before isolation to identify any existing C2 channels or lateral movement targets.

Step 2: Detection — Query SIEM and EDR for anomalous process execution spawned from the ActiveMQ broker JVM process (e.g., activemq.jar or java processes spawning shell interpreters: bash, sh, cmd.exe, powershell.exe). Review web/application server logs for unexpected HTTP POST or GET requests to /api/jolokia or /jolokia paths with exec or write operation types. Hunt for T1190 indicators: unusual outbound connections from ActiveMQ hosts, new scheduled tasks or cron entries, and lateral movement from broker hosts (T1021). Check for CVE-2023-46604 IOC patterns (OpenWire port 61616) as a baseline comparison for actor TTPs.

NIST Phase: Detection Analysis

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

Controls: NIST IR-5 (Incident Monitoring), 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), MITRE ATT&CK T1190 (Exploit Public-Facing Application), MITRE ATT&CK T1021 (Remote Services — Lateral Movement)

Compensating: Deploy Sysmon (config with SwiftOnSecurity or olafhartong template) on all ActiveMQ broker hosts and filter Sysmon Event ID 1 (Process Creation) for ParentImage containing 'java.exe' or 'javaw.exe' with ChildImage matching 'cmd.exe', 'powershell.exe', 'bash', 'sh', or 'curl'. Parse the Jetty access log at $ACTIVEMQ_HOME/data/activemq.log and the web console access log (default: $ACTIVEMQ_HOME/data/audit.log) using grep: 'grep -E "(POST|GET).*/jolokia.*(exec|write|search|list)" /opt/activemq/data/activemq.log'. For CVE-2023-46604 TTP baseline, use Wireshark or tcpdump to capture OpenWire traffic on port 61616 and look for ClassInfo opcodes (0x1f) that reference remote ClassPathXmlApplicationContext URLs — the same actor tradecraft applies to Jolokia-based RCE follow-on. Use the Sigma rule 'proc_creation_java_spawning_shell' (available in SigmaHQ repository) converted to your log platform.

Evidence: Collect the Jetty HTTP access log ($ACTIVEMQ_HOME/data/ or configured log path) covering the 30-day window prior to discovery — Jolokia exploitation leaves HTTP POST requests to /api/jolokia/exec/ with MBean operation names such as 'java.lang:type=Runtime' executeCommand or 'com.sun.management:type=DiagnosticCommand'. Preserve Sysmon Event ID 1 logs or OS audit logs showing java.exe/activemq process ancestry chains. Capture any files dropped in $ACTIVEMQ_HOME/webapps/ (webshells), /tmp/, %TEMP%, or cron.d/crontabs for new entries post-broker start time. For CVE-2024-32114 (authentication bypass on 6.0.0–6.1.1), check broker audit logs for unauthenticated API calls — these will appear as successful 200-response Jolokia operations with no Authorization header in the Jetty access log. Collect Windows Security Event Log Event ID 4688 (Process Creation with command line) or Linux /var/log/auth.log and auditd logs for shell spawns

from the ActiveMQ service account UID.

Step 3: Eradication — Upgrade Apache ActiveMQ Classic to version 5.19.4 (5.x branch) or 6.2.2 (6.x branch) per the Apache ActiveMQ project release page. If immediate patching is not possible: disable the Jolokia endpoint entirely in activemq.xml or jetty.xml by removing or commenting out the Jolokia servlet configuration; enforce authentication on the web console; rotate any default or weak credentials on the broker and management interfaces (address CWE-1392).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality), NIST IA-5 (Authenticator Management), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 5.2 (Use Unique Passwords)

Compensating: If the host cannot be taken offline for patching, apply the Jolokia servlet removal as an emergency configuration change: in $ACTIVEMQ_HOME/conf/jetty.xml, locate and remove or comment out the bean definition referencing 'org.jolokia' or the servlet mapping for '/api/jolokia/*'. Restart the broker service and verify: 'curl -v http://localhost:8161/api/jolokia' should return 404 or connection refused. Rotate the admin password in $ACTIVEMQ_HOME/conf/jetty-realm.properties by replacing the default 'admin: admin, admin' entry with a strong randomly generated password (use 'openssl rand -base64 24'). For credential rotation on the broker itself, update $ACTIVEMQ_HOME/conf/activemq.xml broker authentication plugin entries. Document all temporary configuration changes as a tracked exception under your change management process with a defined patch-by date tied to the CISA KEV April 30, 2026 deadline.

Evidence: Before applying the patch or configuration change, take a binary hash of the existing activemq-broker-*.jar and activemq-all-*.jar files ('sha256sum /opt/activemq/lib/activemq-broker-*.jar') and preserve the original jetty.xml, activemq.xml, and jetty-realm.properties under version control or secure evidence storage — these establish the pre-eradication configuration state for forensic comparison and chain-of-custody. Capture a memory dump of the running ActiveMQ JVM process using jmap ('jmap -dump:format=b,file=activemq_heap.hprof ') before shutdown if exploitation is confirmed — heap analysis may reveal injected class objects or deserialized payloads loaded via the Jolokia exec interface. Preserve the full $ACTIVEMQ_HOME/webapps/ directory tree as a forensic copy before overwriting with the patched version.

Step 4: Recovery — After patching, confirm the running version via the ActiveMQ web console or 'activemq --version'. Validate that the /api/jolokia endpoint returns 401/403 or is unreachable from untrusted networks. Re-enable monitoring and confirm no persistence mechanisms (cron, scheduled tasks, new user accounts, webshells in ActiveMQ's web directory) were established during the exploitation window. Review broker configuration for unauthorized changes.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-7 (Least Functionality), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 4.6 (Securely Manage Enterprise Assets and Software), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Verify the patched version with 'activemq --version' and compare against the expected output for 5.19.4 or 6.2.2. Test Jolokia endpoint closure with 'curl -I http://:8161/api/jolokia' from an untrusted VLAN — expect 401, 403, or connection refused. For persistence sweep: on Linux run 'crontab -l -u activemq; ls -la /etc/cron.d/; find $ACTIVEMQ_HOME/webapps -name "*.jsp" -newer $ACTIVEMQ_HOME/lib/activemq-broker-*.jar' to identify webshells dropped after broker startup. On Windows run 'schtasks /query /fo LIST /v | findstr /i activemq' and 'Get-LocalUser | Where-Object {$_.Enabled -eq $true}' to check for new accounts. Use YARA rules targeting common JSP webshell patterns (e.g., the public 'webshells' YARA ruleset from Neo23x0) against the $ACTIVEMQ_HOME/webapps/ directory.

Evidence: After patching, generate a new SHA-256 hash of the updated activemq-broker-*.jar and compare against the Apache ActiveMQ official release checksum published on the Apache downloads page — document the comparison result for the incident record. Capture a clean post-patch 'curl -v http://localhost:8161/api/jolokia' response

(expecting 401/403/404) as evidentiary proof of remediation for regulatory or audit purposes. Preserve the output of the persistence sweep commands (crontab listings, scheduled task exports, new account enumeration, webshell scan results) as timestamped artifacts in the incident case file. If any webshells or unauthorized cron entries are found, treat this as a confirmed post-exploitation persistence event and escalate to full forensic acquisition before proceeding.

Step 5: Post-Incident — Conduct a full inventory of JMX/Jolokia-enabled services across the environment; this attack surface predates CVE-2026-34197 by 13 years and may affect other products. Evaluate whether management interfaces (ActiveMQ web console, Jolokia, JMX RMI) are appropriately segmented from production and internet-facing networks. Add ActiveMQ version monitoring to your vulnerability management program. Map findings to NIST CSF ID.AM (asset management) and PR.AC (access control) and document control gaps for the next GRC review cycle.

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 SI-5 (Security Alerts, Advisories, and Directives), NIST CM-7 (Least Functionality), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Use osquery to enumerate JMX/Jolokia exposure across the fleet: 'SELECT name, path, pid FROM processes WHERE name LIKE "%java%";' combined with 'SELECT local_port, remote_address FROM process_open_sockets WHERE local_port IN (8161, 1099, 11099);' to find all Java processes with Jolokia (8161) or JMX RMI (1099/11099) ports open. Extend the search beyond ActiveMQ to other Java middleware (Kafka, Elasticsearch, Tomcat, Spring Boot Actuator with /jolokia endpoint) that may embed Jolokia as a library dependency — run 'find / -name "jolokia*.jar" 2>/dev/null' across managed hosts. Subscribe to the Apache Security mailing list (security@apache.org announcements) and configure a CISA KEV RSS feed alert for future ActiveMQ and JMX-related advisories. Document the 13-year JMX exposure window in the lessons-learned report as a systemic control gap in software inventory and management interface segmentation.

Evidence: Compile the full incident timeline from log evidence preserved in Steps 1–4 — specifically the Jetty HTTP access log entries showing Jolokia exec calls, the Sysmon/auditd process creation events, any persistence artifacts found, and the pre/post-patch JAR checksums — into a structured incident report for the GRC review cycle. Produce a network topology diagram identifying which ActiveMQ broker ports (8161, 61616, 1099) were reachable from untrusted segments during the exploitation window; this serves as documented evidence of the PR.AC control gap for the CSF mapping. Retain all forensic artifacts (heap dump, log archives, configuration snapshots) for a minimum of 12 months or per your documented retention policy under NIST AU-11 (Audit Record Retention), as regulatory breach notification obligations (if PII/PHI transited the compromised broker) may require evidence preservation beyond the immediate incident window.

Detection Guidance

Primary indicators: HTTP requests to /api/jolokia or /jolokia with operation types exec, write, or set in the request body, look for these in ActiveMQ access logs and any WAF/proxy logs covering port 8161. Process telemetry: child processes (bash, sh, cmd.exe, powershell.exe, curl, wget) spawned directly from the Java broker process are high-confidence indicators of post-exploitation. Network telemetry: outbound connections from ActiveMQ hosts to non-standard destinations, particularly on non-broker ports, may indicate C2 or exfiltration (T1041, T1071.001). File system: new cron entries, scheduled tasks, or files written to the ActiveMQ web deployment directory (webapps/) warrant immediate investigation. For the chained unauthenticated path (CVE-2024-32114 + CVE-2026-34197 on 6.0.0-6.1.1): any successful Jolokia operation from an unauthenticated source should be treated as confirmed exploitation. Exclude traffic from authorized security scanning and testing tools in your SIEM rules to avoid alert fatigue. Coordinate with your security and development teams to ensure Jolokia

access is logged and monitored separately from general application traffic. Recommended log sources: ActiveMQ access.log, OS auditd or Windows Security Event Log (process creation events 4688/Sysmon Event ID 1), EDR process tree telemetry, network flow logs from broker host segments.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`/api/jolokia (HTTP POST/GET with exec or write operations)`	Jolokia API abuse path used in CVE-2026-34197 exploitation; look for this in ActiveMQ access logs on port 8161	HIGH
URL	`/jolokia (alternate path)`	Alternate Jolokia endpoint path; same detection logic applies	HIGH

Framework Mappings

MITRE-ATTACK

- **T1078.001** — Default Accounts
- **T1021** — Remote Services
- **T1071.001** — Web Protocols
- **T1071** — Application Layer Protocol
- **T1059** — Command and Scripting Interpreter
- **T1041** — Exfiltration Over C2 Channel
- **T1210** — Exploitation of Remote Services
- **T1190** — Exploit Public-Facing Application

NIST-800-53R5

- **AC-17** — Remote Access
- **AC-3** — Access Enforcement
- **CM-7** — Least Functionality
- **IA-2** — Identification and Authentication (Organizational Users)
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-4** — System Monitoring
- **SI-3** — Malicious Code Protection
- **SI-7** — Software, Firmware, and Information Integrity
- **AC-6** — Least Privilege
- **SI-2** — Flaw Remediation
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning

- **SI-10** — Information Input Validation
- **IR-5** — Incident Monitoring

OWASP-TOP10-2021

- **A03:2021** — Injection
- **A07:2021** — Identification and Authentication Failures

CIS-V8

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

ISO-27001-2022

- **A.8.26** — Application security requirements
- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information

HIPAA-SECURITY

- **164.308(a)(6)(ii)** — Response and Reporting

SOC2-TSC

- **CC7.4** — Responds to identified security incidents

NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1078.001	Default Accounts	Defense-Evasion
T1021	Remote Services	Lateral-Movement
T1071.001	Web Protocols	Command-And-Control
T1071	Application Layer Protocol	Command-And-Control
T1059	Command and Scripting Interpreter	Execution
T1041	Exfiltration Over C2 Channel	Exfiltration
T1210	Exploitation of Remote Services	Lateral-Movement
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/04/apache-activemq-cve-2026-34197-ad.. .	T3
CVE-2026-34197 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-34197	T1
CVE-2026-34197 ActiveMQ RCE via Jolokia API | Horizon3.ai	https://horizon3.ai/attack-research/disclosures/cve-2026-34197-acti...	T3
CVE-2026-34197 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-34197	T3
CVE-2026-34197: ActiveMQ RCE Critical Analysis | PurpleOps	https://purple-ops.io/blog/cve-2026-34197-activemq-rce	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-34197, CVE-2024-32114, CV...	T1

Generated 2026-04-17 06:49 UTC by TJS Security Command Center