INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-13 16:27 UTC

mjdm majordomo - mjdm majordomo Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')

CVE VULNERABILITY | CRITICAL | CVSS 9.8 | CISA KEV

SCC Item ID	SCC-CVE-2026-0032
Type	CVE Vulnerability
CVE ID	CVE-2026-27175
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.2519 (96th percentile)
KEV Status	Yes — CISA Known Exploited Vulnerability
Affected Products	MajorDoMo (aka Major Domestic Module), specific version(s) not confirmed in available data
Published	2026-04-13T00:00:00Z
Discovery Source	Vulncheck Kev

Executive Summary

A critical unauthenticated remote code execution vulnerability (CVE-2026-27175, CVSS 9.8) affects MajorDoMo, an open-source smart home automation platform. An attacker with network access to the web interface can execute arbitrary operating system commands on the host without providing any credentials, gaining full control of the underlying system. CISA has added this to the Known Exploited Vulnerabilities catalog, and active exploitation is confirmed; organizations running MajorDoMo must treat this as an emergency.

Technical Analysis

CVE-2026-27175 is an unauthenticated OS command injection vulnerability (CWE-78) in MajorDoMo (Major Domestic Module). The flaw resides in rc/index.php, where user-supplied input in the $param variable is interpolated directly into a shell command string enclosed in double quotes, with no call to escapeshellarg() or equivalent sanitization. The function safe_exec() inserts the unsanitized command into a database-backed execution queue without performing any input validation despite its name. The second component, cycle_execs.php, is web-accessible without authentication and dequeues commands, passing them directly to PHP's exec(). Exploitation requires timing a race condition: an attacker first requests cycle_execs.php to trigger

the polling loop, then submits a malicious payload via the rc endpoint before the poll cycle exits. Shell metacharacters within double quotes expand at execution time, achieving remote code execution typically within one second. No authentication is required at any stage. MITRE ATT&CK techniques: T1190 (Exploit Public-Facing Application) for initial access; T1059.004 (Unix Shell) for execution. Patch availability should be verified directly against the MajorDoMo project repository (https://github.com/sergejey/majordomo, verify this is the official repository) and vendor advisory. EPSS score: 0.252 (96th percentile), indicating high likelihood of active or imminent exploitation. CISA KEV listing confirms in-the-wild exploitation.

Action Checklist

1. Step 1: Containment, Immediately restrict network access to MajorDoMo web interfaces (rc/index.php and cycle_execs.php). Block inbound HTTP/HTTPS to these endpoints at the perimeter firewall or WAF for all internet-facing instances. If the platform is internally hosted, enforce network segmentation to limit lateral access from untrusted segments. Treat any externally accessible MajorDoMo instance as compromised until assessed.
2. Step 2: Detection, Review web server access logs (Apache/Nginx access.log) for requests to rc/index.php with unusual $param values containing shell metacharacters ($, `, ;, |, &&, >). Also audit requests to cycle_execs.php originating from external or unexpected source IPs. Check system process logs for unexpected child processes spawned by the web server user (on Linux: /var/log/auth.log, /var/log/syslog, auditd; on Windows: Event Viewer Application/Security logs). Correlate with EPSS and KEV status to prioritize log review timeframe back to the CVE publication date.
3. Step 3: Eradication, Check the MajorDoMo project repository (https://github.com/sergejey/majordomo, verify this is the official repository) for a vendor-issued patch. If a confirmed patch is available, apply immediately. If no patch is confirmed available, implement compensating controls: (a) restrict access to rc/index.php and cycle_execs.php via .htaccess or web server configuration requiring authentication; (b) add input validation or WAF rules blocking shell metacharacters in the $param parameter. Monitor the project repository and NVD entry (https://nvd.nist.gov/vuln/detail/CVE-2026-27175) for official patch release and apply immediately upon availability.
4. Step 4: Recovery, After applying patch or compensating controls, validate that rc/index.php enforces input sanitization via escapeshellarg() or equivalent, and that cycle_execs.php is no longer accessible without authentication. Conduct a post-compromise review of the host for unauthorized accounts, scheduled tasks (cron on Linux, Task Scheduler on Windows), webshells, or persistence mechanisms. Monitor outbound network connections from the MajorDoMo host for anomalous destinations. Restore from a known-good backup if compromise evidence is found.
5. Step 5: Post-Incident, Document the control gap: externally accessible administrative endpoints with no authentication and no input sanitization in a command execution path. Implement a recurring audit of web-accessible endpoints on internal platforms to confirm authentication requirements. Add CWE-78 (OS Command Injection) to secure code review checklists for any internally developed integrations. Review whether MajorDoMo (or similar IoT/smart home platforms) should be network-isolated by policy regardless of patch status.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to senior IR leadership and legal/compliance counsel immediately if forensic evidence confirms unauthorized remote code execution occurred on the MajorDoMo host prior to containment — specifically if `/etc/passwd` shows added accounts, webshells are found in the docroot, outbound connections to non-organizational IPs are confirmed, or the MajorDoMo host has network adjacency to OT/ICS systems, PII datastores, or regulated environments triggering breach notification obligations.
Recovery Notes	After patching or applying compensating controls, validate remediation by sending a benign test request to `rc/index.php?param=test%3Becho%20CVE-2026-27175` from an authorized internal scanner and confirming the server does not execute the echo command (check for absence of command output in the response and absence of a new process in `ps auxf`). Monitor outbound network connections from the MajorDoMo host for a minimum of 30 days post-recovery using `ss -tnp` logging via cron or osquery, focusing on unexpected connections to external IPs that could indicate a dormant C2 implant placed through the RCE prior to containment. Do not return the MajorDoMo instance to production from a backup without first verifying the backup predates the earliest possible exploitation window identified in the access log review.
Forensic Artifacts	Apache/Nginx access.log: POST or GET requests to /rc/index.php and /cycle_execs.php containing URL-encoded shell metacharacters (%3B, %7C, %60, %24, %26%26, %3E) in the 'param' query parameter — the primary artifact of CVE-2026-27175 exploitation attempts. | auditd EXECVE syscall logs (/var/log/audit/audit.log): EXECVE records showing /bin/sh or /bin/bash spawned with a parent PID matching the Apache2, php-fpm, or Nginx worker process — direct evidence of OS command injection execution via the MajorDoMo web interface. | MajorDoMo webroot filesystem (/var/www/majordomo or equivalent): Newly created or recently modified .php files not present in the original MajorDoMo repository commit history, particularly in upload directories or cache paths — indicative of webshells dropped as a second-stage payload after RCE via CVE-2026-27175. | /etc/passwd, /etc/cron.d/*, and crontab -l output for www-data: Unauthorized OS-level user accounts or cron jobs added post-exploitation to maintain persistence after initial RCE — a common attacker follow-on action once unauthenticated shell access is obtained. | Network flow or pcap data (captured via tcpdump -i eth0 -w /tmp/majordomo_capture.pcap): Outbound HTTP/HTTPS, DNS, or raw TCP connections from the MajorDoMo host to non-organizational external IPs following the exploitation window, consistent with wget/curl-based payload staging or reverse shell C2 callbacks initiated through the injected OS command.

Per-Action IR Details

Step 1: Containment — Immediately restrict network access to MajorDoMo web interfaces (rc/index.php and cycle_execs.php). Block inbound HTTP/HTTPS to these endpoints at the perimeter firewall or WAF for all internet-facing instances. If the platform is internally hosted, enforce network segmentation to limit lateral access from untrusted segments. Treat any externally accessible MajorDoMo instance as compromised until assessed.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), NIST AC-3 (Access Enforcement), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Without an enterprise firewall, use iptables to immediately block external access: `iptables -I INPUT -p tcp --dport 80 -m string --string 'rc/index.php' --algo bm -j DROP` and `iptables -I INPUT -p tcp --dport 80 -m string --string 'cycle_execs.php' --algo bm -j DROP`. For HTTPS (port 443), repeat with `--dport 443`. On the MajorDoMo host itself, add deny rules in the Apache/Nginx vhost config for these two paths and reload the service. A 2-person team can execute both steps in under 10 minutes; one person handles the perimeter, the other handles the host-level config.

Evidence: Before blocking, capture a full netstat or `ss -tnp` snapshot from the MajorDoMo host to record any currently established or TIME_WAIT connections to rc/index.php or cycle_execs.php endpoints — active sessions may indicate in-progress exploitation. Also capture `ps auxf` to record any shell processes currently running under the web server user (www-data or equivalent) that could be live RCE sessions spawned via the $param injection vector. Preserve this output as a timestamped text file before firewall rules terminate existing sessions.

Step 2: Detection — Review web server access logs (Apache/Nginx access.log) for requests to rc/index.php with unusual $param values containing shell metacharacters ($, `, ;, |, &&, >). Also audit requests to cycle_execs.php originating from external or unexpected source IPs. Check system process logs (auditd, /var/log/auth.log, /var/log/syslog) for unexpected child processes spawned by the web server user (e.g., www-data executing shell commands). Correlate with EPSS and KEV status to prioritize log review timeframe back to the CVE publication date.

NIST Phase: Detection Analysis

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

Controls: NIST AU-2 (Event Logging), 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: Run this grep against Apache/Nginx access logs to surface CVE-2026-27175 exploitation attempts targeting the $param OS injection vector: `grep -E 'rc/index\.php|cycle_execs\.php' /var/log/apache2/access.log | grep -E '(%24|\$|%60|`|%3B|;|%7C|\||%26%26|&&|%3E|>)' > /tmp/majordomo_hits.txt`. For process-level evidence, if auditd is running use `ausearch -k execve --start $(date -d 'CVE publication date' '+%m/%d/%Y %H:%M:%S') | grep -A5 'www-data'` to find shell commands spawned by the web server process. Without auditd, install Sysmon for Linux (or use existing Sysmon on Windows hosts) and check for process creation events where the parent PID maps to Apache/Nginx/PHP-FPM. Cross-reference source IPs from the grep output against threat intel feeds using a free tool like GreyNoise Community API.

Evidence: Primary: Apache or Nginx access.log entries for POST/GET requests to `/rc/index.php` and `/cycle_execs.php` containing URL-encoded shell metacharacters in the `param` query parameter (e.g., %3B for `;`, %7C for `|`, %60 for backtick). Secondary: auditd EXECVE syscall records showing `/bin/sh`, `/bin/bash`, `curl`, `wget`, or `python` spawned with PPID matching the web server process (Apache2, php-fpm). Tertiary: `/var/log/auth.log` entries showing new user creation, `sudo` invocations, or SSH key additions under the www-data UID following a suspicious request timestamp. Capture all log files with `md5sum` and `sha256sum` hashes immediately to preserve forensic integrity per NIST 800-61r3 §3.2 evidence handling guidance.

Step 3: Eradication — Apply the vendor-issued patch once confirmed available from the MajorDoMo project repository (https://github.com/sergejey/majordomo). If no patch is confirmed available, implement compensating controls: (a) restrict access to rc/index.php and cycle_execs.php via .htaccess or web server configuration requiring authentication; (b) add input validation or WAF rules blocking shell metacharacters in the $param parameter. Monitor the project repository and NVD entry (https://nvd.nist.gov/vuln/detail/CVE-2026-27175) for official patch release and apply immediately upon availability.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST SI-10 (Information Input Validation), 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 7.4 (Perform Automated Application Patch Management)

Compensating: Until an official patch is released on the MajorDoMo GitHub repository (https://github.com/sergejey/majordomo), add the following to the Apache vhost or `.htaccess` for the MajorDoMo docroot to require HTTP Basic Auth on the two vulnerable endpoints: ` AuthType Basic AuthName 'Restricted' AuthUserFile /etc/apache2/.htpasswd Require valid-user `. Repeat the block for `cycle_execs.php`. Additionally, deploy a ModSecurity rule (free, Apache/Nginx): `SecRule ARGS:param '@rx [;|`$&><]' 'id:9001,phase:2,deny,status:403,msg:CVE-2026-27175 OS Injection Attempt'`. A Sigma rule detecting child process

spawning from the web server process can be deployed via `grep`-based log monitoring if no SIEM is available.

Evidence: Before applying the patch or compensating controls, capture the current state of `rc/index.php` and `cycle_execs.php` source files with SHA-256 hashes (`sha256sum /path/to/majordomo/rc/index.php`) to confirm whether the files have been tampered with by a prior attacker who may have embedded a webshell or modified the injection point to persist access. Also run `find /var/www/majordomo -name '*.php' -newer /var/www/majordomo/index.php -mtime -30` to identify any recently modified or newly created PHP files that may represent dropped webshells introduced through the CVE-2026-27175 RCE path prior to containment.

Step 4: Recovery — After applying patch or compensating controls, validate that rc/index.php enforces input sanitization via escapeshellarg() or equivalent, and that cycle_execs.php is no longer accessible without authentication. Conduct a post-compromise review of the host for unauthorized accounts, scheduled tasks (cron), webshells, or persistence mechanisms. Monitor outbound network connections from the MajorDoMo host for anomalous destinations. Restore from a known-good backup if compromise evidence is found.

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 AC-2 (Account Management), NIST CP-10 (System Recovery and Reconstitution), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 5.3 (Disable Dormant Accounts)

Compensating: Run a targeted persistence hunt with the following commands on the MajorDoMo host: (1) `crontab -l -u www-data` and `cat /etc/cron*/*` to check for attacker-planted cron jobs using the web server user. (2) `grep -rn 'eval(base64_decode' /var/www/majordomo/` and `grep -rn 'system($_' /var/www/majordomo/` to detect PHP webshells dropped via the RCE vector. (3) `awk -F: '($3 >= 1000) {print}' /etc/passwd` to enumerate non-system accounts added post-exploitation. (4) Use `ss -tnp` and `netstat -anp` to identify unexpected outbound connections to C2 infrastructure. For ongoing monitoring without EDR, deploy osquery with a query pack that monitors `/etc/passwd` changes and new cron entries every 5 minutes.

Evidence: Capture the following before restoring from backup to preserve forensic evidence of the full compromise chain: `/tmp/` and `/var/tmp/` directory listings (common attacker staging directories for downloaded payloads delivered via `wget` or `curl` through the RCE), `/root/.bash_history` and `/var/www/.bash_history` for command history reflecting post-exploitation activity, `/etc/passwd` and `/etc/shadow` for unauthorized account additions, and a full recursive listing of the MajorDoMo webroot with timestamps (`find /var/www/majordomo -printf '%T+ %p ' | sort > /tmp/webroot_timestamps.txt`) to identify files created or modified after the earliest confirmed malicious request in the access logs.

Step 5: Post-Incident — Document the control gap: externally accessible administrative endpoints with no authentication and no input sanitization in a command execution path. Implement a recurring audit of web-accessible endpoints on internal platforms to confirm authentication requirements. Add CWE-78 (OS Command Injection) to secure code review checklists for any internally developed integrations. Review whether MajorDoMo (or similar IoT/smart home platforms) should be network-isolated by policy regardless of patch status.

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-2 (Flaw Remediation), NIST SI-10 (Information Input Validation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 4.2 (Establish and Maintain a Secure Configuration Process for Network Infrastructure)

Compensating: Without a formal vulnerability management platform, implement a monthly recurring task (cron job or calendar reminder) to run `curl -s 'https://services.nvd.nist.gov/rest/json/cves/2.0?keywordSearch=majordomo' | python3 -m json.tool | grep cvssV3` against the NVD API to catch new CVEs affecting MajorDoMo and similar open-source IoT platforms. Add a YARA rule to your next code review cycle targeting CWE-78 patterns in PHP: `rule CWE78_OSCommandInjection { strings: $s1 = "exec(" $s2 = "shell_exec(" $s3 = "passthru(" $s4 = "system(" $s5 = "popen(" condition: any of them }` applied to any internally developed MajorDoMo module or integration. Document the network isolation policy decision for IoT/smart home platforms in the risk register regardless of patch status.

Evidence: For the lessons-learned documentation, preserve the full timeline reconstruction from Apache access logs showing the earliest possible exploitation window (first malicious request to rc/index.php or cycle_execs.php with shell metacharacters in $param) through to containment, including all source IPs, User-Agent strings, and payload patterns observed. This timeline supports both internal reporting and any required regulatory notification, and establishes the basis for updating detection rules and WAF signatures to catch CVE-2026-27175 exploitation patterns against MajorDoMo in any future redeployment.

Detection Guidance

Primary indicators: POST or GET requests to /rc/index.php containing shell metacharacters in the param field ($(), `, ;, |, &&, ||, >). Requests to /cycle_execs.php from external or unexpected source IPs, particularly in rapid succession with rc/index.php requests (within 1-2 seconds). Behavioral indicators: web server process (www-data or equivalent) spawning unexpected child processes such as bash, sh, or nc (netcat), tools not typically part of MajorDoMo's normal operation. Curl and wget may be legitimate depending on the environment; correlate with suspicious rc/index.php requests before alerting. Additional indicators: new cron entries (Linux) or scheduled tasks (Windows), SSH authorized_keys modifications, or new user accounts created around the time of suspicious web requests. Log sources to query: web server access logs (filter on rc/index.php and cycle_execs.php URIs), on Linux, auditd syscall logs (execve calls by web server UID), on Windows, Event Viewer for process creation events by the web server process, and database logs for unexpected insertions into the command queue table. If a SIEM is available, correlate: rc/index.php hit followed within 2 seconds by a cycle_execs.php hit from the same source IP, combined with a new process spawned by the web server user.

Framework Mappings

MITRE-ATTACK

- **T1059.004** — Unix Shell
- **T1190** — Exploit Public-Facing Application

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **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
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A03:2021** — Injection

CIS-V8

- **2.5** — Allowlist Authorized Software

- **16.10** — Apply Secure Design Principles in Application Architectures

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1059.004	Unix Shell	Execution
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
vulncheck_kev	https://nvd.nist.gov/vuln/detail/CVE-2026-27175	T1
CVE-2026-27175: MajorDoMo OS Command Injection RCE Flaw	https://www.sentinelone.com/vulnerability-database/cve-2026-27175/	T3
CVE-2026-27175 - Exploits & Severity - Feedly	https://feedly.com/cve/CVE-2026-27175	T3
CVE-2026-27175 - Vulnerability-Lookup	https://db.gcve.eu/vuln/cve-2026-27175	T3
CVE-2026-27175 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-27175	T3
CISA KEV	https://www.cisa.gov/known-exploited-vulnerabilities-catalog	T1

Generated 2026-04-13 16:27 UTC by TJS Security Command Center