INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-18 13:46 UTC

mjdm majordomo - mjdm majordomo Improper Control of Generation of Code ('Code Injection')

CVE VULNERABILITY | CRITICAL | CVSS 9.8 | CISA KEV

SCC Item ID	SCC-CVE-2026-0051
Type	CVE Vulnerability
CVE ID	CVE-2026-27174
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.5163 (98th percentile)
KEV Status	Yes — CISA Known Exploited Vulnerability
Affected Products	MajorDoMo (aka Major Domestic Module), version unspecified; admin panel with PHP console feature enabled
Published	2026-04-18T00:00:00Z
Discovery Source	Vulncheck Kev

Executive Summary

A critical unauthenticated remote code execution vulnerability in MajorDoMo, an open-source home automation platform, allows any attacker to run arbitrary commands on affected systems without credentials. The vulnerability stems from a missing exit statement after a redirect call and direct use of user-supplied input in PHP's eval() function. CISA has confirmed active exploitation (see CISA Known Exploited Vulnerabilities Catalog); organizations running MajorDoMo with the admin panel PHP console enabled should treat this as an immediate containment priority.

Technical Analysis

CVE-2026-27174 (CVSS 9.8 / Critical) affects MajorDoMo (Major Domestic Module) in all versions with the admin panel PHP console feature enabled. The root cause is a missing exit statement following a redirect() call in modules/panel.class.php (CWE-670), which allows unauthenticated requests to fall through to the ajax handler in inc_panel_ajax.php. That handler passes user-controlled GET parameters, delivered via register_globals, directly to PHP's eval() without any authentication check (CWE-306, CWE-94). An attacker sends a crafted GET request to /admin.php with parameters ajax_panel, op, and command to execute arbitrary PHP code. MITRE techniques: T1190 (Exploit Public-Facing Application), T1203 (Exploitation for Client Execution), T1059.004 (Unix Shell). A public proof-of-concept exists on GitHub. Confirmed in both CISA KEV

and VulnCheck KEV. No vendor CVSS vector was provided as of the publication date. EPSS score is 0.5163 (97.9th percentile), indicating very high exploitation likelihood in the near term. No patch version has been released as of publication date; monitor the MajorDoMo project repository for updates.

Action Checklist

1. Step 1: Containment. Immediately disable or block access to the MajorDoMo admin panel (/admin.php) at the network perimeter. If the system is internet-facing, apply firewall or WAF rules to block all external access to the admin panel endpoint. Isolate affected hosts from broader network segments until remediation is confirmed.
2. Step 2: Detection. Search web server and application logs for GET requests to /admin.php containing the parameters ajax_panel, op, and command. Flag any requests matching this pattern regardless of response code. Note: a 302 redirect response does not indicate failed exploitation, the include order bug allows execution to continue after the redirect. Review for evidence of outbound connections, new processes spawned by the web server user, or unexpected file writes following such requests. Check CISA KEV catalog and internal SIEM for IOC matches against this CVE.
3. Step 3: Eradication. Disable the PHP console feature within the MajorDoMo admin panel configuration, or take the system offline entirely. Monitor the MajorDoMo project repository and vendor advisory channel for a patched release; apply when available. If a patched version is not yet available, do not attempt code modifications unless your team has PHP security expertise.
4. Step 4: Recovery. After applying mitigations, verify that /admin.php no longer responds to unauthenticated requests containing console parameters. Re-scan the host for web shells, unauthorized files, or scheduled tasks created during any exploitation window. Restore from a known-good backup if compromise is confirmed. Monitor outbound traffic from the host for 30 days post-remediation.
5. Step 5: Post-Incident. This vulnerability exposes two persistent control gaps: absence of authentication enforcement on internal code execution paths, and use of register_globals with eval() on a public-facing panel. Review all other admin panel features for similar authentication bypass patterns. Assess whether MajorDoMo deployments require internet exposure at all; restrict to internal network or VPN access as a standing policy. Add detection rules for eval()-triggered process spawning on PHP application servers.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to senior IR leadership, legal, and privacy counsel immediately if forensic analysis confirms successful exploitation (HTTP 200 responses to console parameter requests, evidence of web shell deployment, or outbound connections from the web server process), as CVE-2026-27174 is CISA KEV-confirmed with active exploitation, CVSS 9.8, and unauthenticated RCE on a home automation platform that may have access to physical building systems, IoT devices, or networks containing PII/PHI subject to breach notification requirements.

Recovery Notes	Before returning the MajorDoMo host to production, verify the PHP console eval() code path is fully disabled or patched and confirm with an authenticated test request that unauthenticated console parameter submissions return no code execution. If compromise is confirmed, rebuild from a known-good image rather than attempting in-place cleanup, as MajorDoMo's integration with physical home automation devices (smart locks, cameras, HVAC) means a compromised host may have been used to alter device configurations or automation rules that persist independently of the web application. Maintain 30-day enhanced monitoring of outbound traffic from the recovered host, specifically watching for beaconing patterns on non-standard ports, DNS requests to newly registered domains, or connections to IP ranges not associated with MajorDoMo's legitimate cloud services.
Forensic Artifacts	Web server access logs (Apache /var/log/apache2/access.log or Nginx /var/log/nginx/access.log): Preserved for GET/POST requests to /admin.php containing the specific parameter chain 'ajax_panel', 'op', and 'command' — the exact exploit path for CVE-2026-27174's unauthenticated eval() injection. | PHP-spawned child process records: Auditd EXECVE syscall records (ausearch -sc execve) or Sysmon Event ID 1 entries showing processes with a PHP interpreter parent (php, php-fpm, php-cgi) — direct forensic evidence of successful arbitrary command execution via the eval() sink in the MajorDoMo console module. | Web root filesystem timeline (/var/www/majordomo/): Files created or modified after the first suspicious /admin.php request, particularly .php files in world-writable directories or /tmp, consistent with attacker-deployed web shells leveraging the unauthenticated RCE to establish persistence. | MajorDoMo automation database (typically SQLite or MySQL, referenced in MajorDoMo config.php): Attacker-modified automation rules or newly created device scripts within the home automation logic layer, which an attacker with RCE access could alter to maintain persistent access or pivot to connected IoT devices. | Outbound network connection records (netflow, iptables LOG rules, or tcpdump captures): Connections originating from the web server process user (www-data, apache) to external IPs on non-standard ports in the minutes following exploitation-pattern requests to /admin.php, indicative of reverse shell callback or data exfiltration initiated through the eval() RCE.

Per-Action IR Details

Step 1: Containment — Immediately disable or block access to the MajorDoMo admin panel (/admin.php) at the network perimeter. If the system is internet-facing, apply firewall or WAF rules to block all external access to the admin panel endpoint. Isolate affected hosts from broader network segments until remediation is confirmed.

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: On Linux hosts running MajorDoMo, immediately apply an iptables rule to drop inbound traffic on port 80/443 destined for /admin.php: `iptables -I INPUT -p tcp --dport 80 -m string --string '/admin.php' --algo bm -j DROP`. On Windows, use Windows Firewall (netsh) to block the port entirely if the admin panel cannot be path-filtered. Alternatively, rename or chmod 000 the /admin.php file at the filesystem level as an emergency measure while a proper WAF rule is deployed. A two-person team can execute this in under 5 minutes without SIEM or EDR.

Evidence: Before blocking, capture the full web server access log (Apache: /var/log/apache2/access.log or /var/log/httpd/access_log; Nginx: /var/log/nginx/access.log) and preserve a timestamped copy to immutable storage. Record active network connections from the MajorDoMo host using `ss -tunap` or `netstat -anp` to capture any established outbound sessions initiated by the web server process (www-data, apache, nginx) that may indicate a live reverse shell or C2 callback established via the eval() injection prior to containment. Preserve the MajorDoMo application log if enabled (typically under the MajorDoMo install directory, e.g., /var/www/majordomo/log/).

Step 2: Detection — Search web server and application logs for GET requests to /admin.php containing the parameters ajax_panel, op, and command. Flag any requests matching this pattern regardless of response code. Review for evidence of outbound connections, new processes spawned by the web server user, or unexpected file writes following such requests. Check CISA KEV catalog and internal SIEM for IOC matches against this CVE.

NIST Phase: Detection Analysis

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

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-3 (Content of Audit Records), CIS 8.2 (Collect Audit Logs)

Compensating: Run this grep against the web server access log to identify exploitation attempts against the MajorDoMo PHP console: `grep -E '/admin\.php.*ajax_panel.*op.*command|/admin\.php.*command=.*eval' /var/log/apache2/access.log`. For process spawning evidence on Linux, search auditd logs (`ausearch -c php -i`) or, if auditd is not running, check /proc for orphaned processes owned by www-data with parent PID of the web server. On Windows hosts, query Sysmon Event ID 1 (Process Creation) filtering for processes where ParentImage matches the PHP interpreter (php.exe or php-cgi.exe). Deploy the public Sigma rule for PHP webshell process spawning (sigma rule id: e7ec56d8-d9b8-4e4d-a0e9-cbd0e46f0ab3 or equivalent) against local logs using sigmac converted to grep. Use osquery to query `SELECT * FROM processes WHERE parent IN (SELECT pid FROM processes WHERE name LIKE '%php%')` to identify child processes spawned by PHP at the time of suspicious requests.

Evidence: Extract and preserve web server access log lines matching GET/POST to /admin.php with parameters ajax_panel, op, and command — the specific parameter chain that the CVE-2026-27174 exploit traverses to reach the eval() sink. Capture HTTP response codes for each matching request (200 responses confirm the PHP console was reached; non-200 does not rule out exploitation due to the missing exit-after-redirect flaw). Collect filesystem modification timestamps (`find /var/www/majordomo -newer /var/log/apache2/access.log -type f`) to identify files written by the web server user after the first suspicious request. Preserve /tmp and the web root for any newly created .php files consistent with web shell deployment (e.g., files with eval(), base64_decode(), or system() calls). Review /etc/crontab and crontabs for www-data for any entries created after the earliest suspicious log timestamp.

Step 3: Eradication — Disable the PHP console feature within the MajorDoMo admin panel configuration if the application cannot be taken offline. Monitor the MajorDoMo project repository and vendor advisory channel for a patched release; apply when available. If a patched version is not yet available, consider replacing eval()-based console functionality with a sandboxed alternative or removing the module entirely.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality), NIST SI-10 (Information Input Validation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Until an official patch is released by the MajorDoMo project (monitor https://github.com/sergejey/majordomo for commits addressing the eval() console), manually edit the MajorDoMo source file that implements the PHP console (typically within the admin panel module, search for `eval(` in /var/www/majordomo/modules/ using `grep -rn 'eval(' /var/www/majordomo/`). Comment out or replace the eval() call with a function that returns an error. Alternatively, remove execute permissions from the console module file: `chmod 000 /var/www/majordomo/modules/.php`. Add a PHP-level authentication check (session token verification) at the top of admin.php as an interim patch if source modification is within team capability. Deploy a YARA rule scanning the web root for eval()-containing PHP files modified within the incident window to detect any attacker-planted persistence: `yara -r eval_webshell.yar /var/www/majordomo/`.

Evidence: Before modifying any application files, image the MajorDoMo web root directory (`tar -czf majordomo_webroot_$(date +%Y%m%d%H%M%S).tar.gz /var/www/majordomo/`) to preserve forensic state. Document the exact eval() code path in admin.php that lacks the exit statement after the redirect — this is the specific flaw in CVE-2026-27174 and must be preserved as evidence of the vulnerability's root cause. Capture the MajorDoMo configuration file (config.php or equivalent) to document whether the PHP console feature was explicitly enabled and

whether any authentication bypass settings (e.g., register_globals equivalents) were active. Preserve any web shells or attacker-dropped files found in the web root before removal, storing them in a password-protected archive for later analysis.

Step 4: Recovery — After applying mitigations, verify that /admin.php no longer responds to unauthenticated requests containing console parameters. Re-scan the host for web shells, unauthorized files, or scheduled tasks created during any exploitation window. Restore from a known-good backup if compromise is confirmed. Monitor outbound traffic from the host for 30 days post-remediation.

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 CP-10 (System Recovery and Reconstitution), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Verify the mitigation is effective by sending a crafted test request (from an authorized test system only, not from the internet): `curl -v 'http:///admin.php?ajax_panel=1&op=console&command=phpinfo()'` — a correctly mitigated system should return a redirect with no eval() execution or a 403/404. Run ClamAV against the web root to detect known web shell signatures: `clamscan -r /var/www/majordomo/ --log=/tmp/clamscan_$(date +%Y%m%d).log`. Audit scheduled tasks using `crontab -l -u www-data` and `ls -la /etc/cron.*` and `systemctl list-timers` for any persistence mechanisms installed during the exploitation window. For 30-day outbound traffic monitoring without a SIEM, configure tcpdump on the host with a daily rotation: `tcpdump -i eth0 -w /captures/majordomo_%Y%m%d.pcap -G 86400 'src host '` and review weekly for unexpected destinations, particularly on non-standard ports consistent with reverse shell or C2 traffic.

Evidence: Before restoring from backup, document all unauthorized files found in the web root, /tmp, and any world-writable directories, including their SHA-256 hashes (`sha256sum `), creation timestamps, and ownership. Capture the current state of the MajorDoMo database if accessible, as attackers may have modified device automation rules or user accounts to establish persistence within the home automation logic layer. Review SSH authorized_keys for www-data and root accounts for any keys added during the exploitation window. Document outbound connection history from the host prior to containment using firewall logs or netflow data to identify any data exfiltration or C2 callback destinations that should be reported or blocked.

Step 5: Post-Incident — This vulnerability exposes two persistent control gaps: absence of authentication enforcement on internal code execution paths, and use of register_globals with eval() on a public-facing panel. Review all other admin panel features for similar authentication bypass patterns. Assess whether MajorDoMo deployments require internet exposure at all; restrict to internal network or VPN access as a standing policy. Add detection rules for eval()-triggered process spawning on PHP application servers.

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-2 (Flaw Remediation), NIST SI-10 (Information Input Validation), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.4 (Require MFA for Remote Network Access), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Write and deploy a Sigma rule targeting PHP eval()-triggered child process spawning for use with any log management tool that accepts Sigma: detection logic should alert on processes where ParentImage contains 'php' AND Image contains common shell binaries (bash, sh, cmd.exe, powershell.exe, wget, curl). Add a persistent iptables rule (saved via iptables-save) restricting /admin.php access to RFC1918 addresses only, enforced at the host level as a standing policy independent of perimeter controls. Conduct a manual code audit of all MajorDoMo modules containing eval(), exec(), system(), shell_exec(), or passthru() calls using: `grep -rn -E 'eval\(|exec\(|system\(|shell_exec\(|passthru\(' /var/www/majordomo/` and document findings in a risk register. Submit IOCs (source IPs observed in exploitation attempts against /admin.php) to CISA's automated indicator sharing (AIS) program to contribute to community defense.

Evidence: Compile a complete incident timeline from web server access logs, auditd logs, and any available netflow data documenting the first observed exploitation attempt against /admin.php (ajax_panel/op/command parameter pattern), any successful code execution events, attacker post-exploitation activity, and the time-to-contain metric. Preserve all forensic artifacts (web root image, log copies, captured files) for a minimum of 12 months per NIST AU-11 (Audit Record Retention) requirements, or longer if regulatory obligations (e.g., HIPAA, PCI-DSS) apply to the environment where MajorDoMo was deployed. Document the specific MajorDoMo version and configuration state (PHP console enabled, authentication configuration) as evidence for the root cause analysis and to inform future procurement/deployment standards for home automation platforms in the organization.

Detection Guidance

Query web server access logs for GET requests to /admin.php where the query string contains ajax_panel, op, and command parameters simultaneously. Example pattern (grep): grep -E 'GET /admin\.php.*ajax_panel.*command' access.log. In a SIEM, alert on any HTTP request to /admin.php with all three parameters present in the URI, regardless of HTTP response code. Note: a 302 redirect response does not indicate failed exploitation, the include order bug allows execution to continue after the redirect. Alert on all matching requests. Monitor for child processes spawned by the web server user (e.g., www-data, apache) that are shells (sh, bash, python, curl, wget). Check for new or modified files in the MajorDoMo web root following any matched log entry. If network monitoring is available, flag outbound connections from the MajorDoMo host to external IPs initiated by the web server process. The public PoC on GitHub (MaxMnMl/majordomo-CVE-2026-27174-poc) documents the exact request structure and can inform signature development.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`/admin.php?ajax_panel=1&op=console&command=`	GET request pattern used to trigger unauthenticated RCE via the MajorDoMo PHP console ajax handler	HIGH

Framework Mappings

MITRE-ATTACK

- **T1203** — Exploitation for Client Execution
- **T1190** — Exploit Public-Facing Application
- **T1059.004** — Unix Shell

NIST-800-53R5

- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-8** — Penetration Testing

- **RA-5** — Vulnerability Monitoring and Scanning
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-7** — Least Functionality
- **SI-10** — Information Input Validation
- **IA-2** — Identification and Authentication (Organizational Users)

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

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
T1203	Exploitation for Client Execution	Execution
T1190	Exploit Public-Facing Application	Initial-Access
T1059.004	Unix Shell	Execution

Sources

Source	URL	Tier
vulncheck_kev	https://nvd.nist.gov/vuln/detail/CVE-2026-27174	T1
Vulnerability Details : CVE-2026-27174 - Majordomo	https://www.cvedetails.com/cve/CVE-2026-27174/	T3
CVE-2026-27174: MajorDoMo RCE Vulnerability - SentinelOne	https://www.sentinelone.com/vulnerability-database/cve-2026-27174/	T3
MaxMnMl/majordomo-CVE-2026-27174-poc - GitHub	https://github.com/MaxMnMl/majordomo-CVE-2026-27174-poc	T3
CVE-2026-27174 - Vulnerability-Lookup	https://db.gcve.eu/vuln/CVE-2026-27174	T3

Source	URL	Tier
CISA KEV	https://www.cisa.gov/known-exploited-vulnerabilities-catalog	T1

Generated 2026-04-18 13:46 UTC by TJS Security Command Center