INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-17 08:04 UTC

CVE-2026-11526: GD versions before 2.86 for Perl allow OS command injection and file overwrite via a 2-arg open() of...

CVE VULNERABILITY | **CRITICAL** | CVSS 9.8

SCC Item ID	SCC-CVE-2026-0316
Type	CVE Vulnerability
CVE ID	CVE-2026-11526
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.0246 (82th percentile)
Affected Products	Perl GD (GD module for Perl) versions before 2.86
Published	2026-06-14T12:16:22.403
Discovery Source	Nvd

Executive Summary

A critical vulnerability in the Perl GD image library (versions before 2.86) allows attackers to execute arbitrary operating system commands or overwrite files on any server running an application that passes unsanitized user-supplied filename strings to GD file-opening constructors. Any web application or backend service built in Perl that uses GD for image processing and accepts user-supplied filenames is potentially exposed. If exploited, an attacker can run commands under the application's system account, enabling data theft, ransomware deployment, or full server compromise.

Technical Analysis

CVE-2026-11526 is a critical-severity (CVSS 9.8) OS command injection and file overwrite vulnerability in the Perl GD image processing module, affecting all versions before 2.86. The root cause is the _make_filehandle internal function, which uses Perl's 2-argument open() form to open file paths passed by callers. Perl's 2-arg open() treats filenames beginning or ending with a pipe character as shell commands (e.g., '| id' or 'id |') and filenames beginning with '>' or '>>' as file write/append redirects. This function is the unified file-opening code path for all filename-accepting constructors: new(), newFromPng(), newFromJpeg(), and similar methods. Any application passing unsanitized, user-controlled strings to these constructors is vulnerable to arbitrary command execution (CWE-78) or file overwrite (CWE-73) under the process's effective UID. In-memory *Data variants that

accept raw data rather than file paths are not affected. No CVSS vendor score or CISA KEV listing is present at this time; CVSS vector pending NVD publication. EPSS score is 0.02459 (82nd percentile), indicating elevated exploitation likelihood relative to the broader CVE population. MITRE ATT&CK techniques: T1190 (Exploit Public-Facing Application), T1059 (Command and Scripting Interpreter), T1059.004 (Unix Shell). CWEs: CWE-73 (External Control of File Name or Path), CWE-78 (Improper Neutralization of Special Elements used in an OS Command). Sources: NVD (T1), CVE Record (cve.org).

Action Checklist

1. Step 1: Containment, Immediately identify all production systems running Perl applications that use the GD module. Query package managers (cpan, cpanm, apt, yum) and Dockerfile/Cartonfile dependency manifests for 'GD' entries with versions below 2.86. For any confirmed instance, enforce input validation at the application layer: reject or sanitize filename arguments containing pipe characters ('|'), leading/trailing whitespace plus pipe, or redirect operators ('>', '>>') before they reach GD constructors. If sanitization cannot be deployed immediately, consider temporarily disabling the image upload or file-processing feature as a containment measure. Apply NIST AC-6 (Least Privilege): confirm the application process runs under the minimum required UID/GID, reducing blast radius if exploitation occurs.
2. Step 2: Detection, Search application logs and web server access logs for requests containing pipe characters or redirect operators in filename parameters submitted to image-processing endpoints (e.g., URL parameters, multipart form fields named 'filename', 'file', 'image', 'path'). In SIEM, query for process creation events spawned by the web server or application user account that are not expected child processes (e.g., /bin/sh, /bin/bash, curl, wget, nc launched by a Perl web process). On Linux, auditd rules targeting execve() calls from the application's effective UID will surface command injection attempts. Review file system audit logs for unexpected file creation or truncation in application-writable directories. Per NIST AU-2 (Event Logging) and AU-6 (Audit Record Review, Analysis, and Reporting), confirm logging is enabled and covers process execution and file access events. Per CIS 8.2, verify audit log collection is active across affected assets.
3. Step 3: Eradication, Upgrade the Perl GD module to version 2.86 or later on all affected systems. Use the appropriate package manager: 'cpan install GD', 'cpanm GD@2.86', or the OS distribution package manager if a patched package is available. Verify the installed version post-upgrade: run 'perl -MGD -e "print GD->VERSION"' and confirm output is 2.86 or higher. For containerized environments, rebuild and redeploy images with the updated GD version and rotate any secrets accessible to the application process as a precautionary measure. Per CIS 7.3 and CIS 7.4, enforce automated patch management to catch future dependency updates. Update baseline configuration documentation to reflect the new GD version requirement.
4. Step 4: Recovery, After upgrading, re-run the version verification command on each host. Conduct a targeted smoke test of all image-processing workflows to confirm application functionality is restored. Monitor application logs and process execution audit logs for 24 to 48 hours post-remediation for any anomalous process spawning or unexpected file writes that could indicate a pre-remediation compromise persisted. Per NIST AU-6 (Audit Record Review, Analysis, and Reporting), schedule an elevated log review frequency during the recovery window. If any evidence of pre-patch exploitation is found, treat the system as compromised and initiate full incident response per your IR playbook.
5. Step 5: Post-Incident, Conduct a dependency review across all Perl-based applications to identify any other use of 2-arg open() patterns in internally developed code, which carries the same risk class. Establish a software composition analysis (SCA) process to track third-party Perl module versions against

known-vulnerable versions, per CIS 2.1 (Establish and Maintain a Software Inventory) and CIS 7.1 (Establish and Maintain a Vulnerability Management Process). Add automated checks for pipe and redirect characters in all filename-accepting API parameters as a defensive coding standard. Document this gap in your risk register and assign ownership for Perl dependency patching cadence.

Detection Guidance

Focus detection on two signals: (1) anomalous process execution and (2) malformed filename input patterns. For process execution, configure auditd or EDR to alert on execve() syscalls where the parent process belongs to the Perl web application user account and the child process is a shell or network utility (e.g., /bin/sh, /bin/bash, curl, wget, python, nc). This pattern indicates successful command injection via the pipe-form open(). For input pattern detection, query WAF or application logs for filename parameters containing pipe characters ('|'), leading or trailing pipes, or redirect operators ('>', '>>'). SIEM SPL example: search sourcetype=access_log (uri_query='*|*' OR uri_query='*>*') | stats count by src_ip, uri_path. For file overwrite detection, monitor inotify or auditd file-watch rules on sensitive directories for unexpected file creation or truncation events owned by the application user. Per NIST AU-3, ensure audit records capture: event type, timestamp, source identity, object affected, and outcome. Per NIST AU-12 (Audit Record Generation), confirm the application generates and forwards these events to your centralized logging platform. Per NIST AU-6 (Audit Record Review), baseline expected process trees for application accounts and alert on deviations. Per NIST SI-7 (Information System Monitoring), monitor system executables and configuration files for unexpected modification following a GD constructor invocation.

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1059** — Command and Scripting Interpreter
- **T1059.004** — Unix Shell

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
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **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

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

MITRE ATT&CK Mapping

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

Sources

Source	URL	Tier
nvd	https://nvd.nist.gov/vuln/detail/CVE-2026-11526	T1
CVE-2026-11526 - Vulnerability Details - OpenCVE	https://app.opencve.io/cve/CVE-2026-11526	T3
CVE-2026-11526 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-11526	T3
CVE-2026-11526: CWE-78 Improper Neutralization of Special ...	https://radar.offseq.com/threat/cve-2026-11526-cwe-78-improper-neut...	T3

Generated 2026-06-17 08:04 UTC by TJS Security Command Center