INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-06 09:04 UTC

GHSA-pwv6-vv43-88gr: Pillow has an OOB Write with Invalid PSD Tile Extents (Integer Overflow)

CVE VULNERABILITY | **HIGH** | CVSS 7.8

SCC Item ID	SCC-CVE-2026-0132
Type	CVE Vulnerability
CVE ID	CVE-2026-42311
Severity	HIGH
CVSS Base Score	7.8
Affected Products	Pillow (PyPI), specific version range not confirmed from available sources
Published	2026-05-04T20:20:31Z
Discovery Source	Osv

Executive Summary

A high-severity integer overflow vulnerability (CVE-2026-42311) in Pillow, a widely used Python imaging library, allows an attacker to trigger an out-of-bounds memory write by supplying a maliciously crafted PSD file. Any application that processes untrusted image uploads using Pillow is potentially exposed to memory corruption, which could lead to application crashes or arbitrary code execution. Organizations running Python-based web services, data pipelines, or content management systems that accept image input should treat this as a priority remediation item.

Technical Analysis

CVE-2026-42311 (GHSA-pwv6-vv43-88gr) is an integer overflow (CWE-190) in Pillow's PSD tile extent parsing logic that leads to an out-of-bounds write (CWE-787). When Pillow processes a PSD file with maliciously crafted tile extent values, the integer overflow corrupts memory bounds calculations, enabling a write operation beyond the allocated buffer. This maps to MITRE ATT&CK T1203 (Exploitation for Client Execution). CVSS base score is 7.8 (NVD published). Affected version range is not yet confirmed from available sources; monitor the Pillow GitHub repository and PyPI release page for patch information. GHSA-pwv6-vv43-88gr on OSV.dev is the authoritative advisory identifier. EPSS score is not yet populated (0.0), indicating limited exploit activity data at time of publication. CISA KEV status: not listed.

Action Checklist

1. Containment: Audit all Python environments and application dependencies for Pillow installations using 'pip list' or dependency scanning tools (e.g., pip-audit, Dependabot). Restrict or disable untrusted PSD file upload functionality in any Pillow-dependent application until a patched version is confirmed and deployed.
2. Detection: Search application logs for PSD file processing activity, particularly large or malformed files. Alert on application crashes or segmentation faults (signal 11) in services that call Pillow image parsing. No specific CVE-linked IOC hashes are available at this time.
3. Eradication: Monitor the Pillow PyPI page (https://pypi.org/project/Pillow/) and the official GitHub repository (https://github.com/python-pillow/Pillow) for a patched release addressing GHSA-pwv6-vv43-88gr. Apply the patched version via 'pip install --upgrade Pillow' once confirmed. Verify via 'pip show Pillow' post-upgrade.
4. Recovery: After upgrading, re-run dependency scans to confirm the patched version is deployed across all environments, including containers and CI/CD pipelines. Test image processing with known-good PSD files. Monitor application error rates for 24-48 hours post-deployment.
5. Post-Incident: Review image input validation controls - enforce file type allowlists, size limits, and consider sandboxed image processing for untrusted input. Evaluate whether software composition analysis (SCA) tooling is integrated into your CI/CD pipeline to catch future library-level CVEs before reaching production.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to security leadership and legal/compliance if web server or application logs confirm PSD uploads from external sources during the vulnerability window, particularly if the application processes user-generated content containing PII or PHI, as successful exploitation of CVE-2026-42311's OOB write could constitute a data breach triggering GDPR Article 33, HIPAA Breach Notification Rule, or applicable state notification obligations.
Recovery Notes	After deploying the patched Pillow version, verify the fix is active in every execution context — virtualenvs, Docker containers, Lambda layers, and CI runners — not just the primary application host, as Pillow is commonly duplicated across isolated Python environments. Re-enable PSD upload functionality only after 'pip-audit' returns clean on all environments and a 24-48 hour post-deployment monitoring window confirms cessation of SIGSEGV/SIGABRT signals and normalized HTTP 5xx error rates on image-processing endpoints. Retain all forensic artifacts (malformed PSD files, crash logs, access log excerpts) in write-protected storage for at least 90 days in the event that post-incident analysis reveals active exploitation requiring breach notification.

Forensic Artifacts	Web server access logs (nginx/Apache) filtered for Content-Type 'image/vnd.adobe.photoshop' or URI patterns matching PSD upload endpoints — capture source IPs, timestamps, response codes, and payload sizes to identify potential exploit delivery attempts against the CVE-2026-42311 PSD tile parsing path | Linux kernel and systemd journal crash records — 'grep -E "segfault|signal 11|python" /var/log/kern.log' and 'journalctl -u | grep -iE "killed|SIGSEGV|SIGABRT|restart"' — an integer overflow triggering OOB write in Pillow's PSD decoder would manifest as a SIGSEGV in the Python worker process | Python application stderr and Gunicorn/uWSGI error logs containing tracebacks referencing Pillow internals, specifically PsdImagePlugin.py or ImagingDecoder, which would appear if the OOB write was partially caught by Python's exception handling before process termination | Preserved copies of all PSD files submitted to the application during the vulnerability window — store write-protected with SHA256 hashes recorded; crafted exploit files targeting CVE-2026-42311 would contain anomalous PSD tile extent values (oversized or negative integer values in the PSD layer and channel header fields) distinguishable from legitimate PSD files | Output of 'pip show Pillow' and 'pip-audit' captured from each affected environment at time of discovery — establishes the vulnerable version, installation path, and dependency chain for the incident record and confirms scope of exposure across virtualenvs, containers, and CI runners

Per-Action IR Details

Containment — Audit all Python environments and application dependencies for Pillow installations using 'pip list' or dependency scanning tools (e.g., pip-audit, Dependabot). Restrict or disable untrusted PSD file upload functionality in any Pillow-dependent application until a patched version is confirmed and deployed.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolate affected components and restrict the attack vector (untrusted PSD ingestion via Pillow) before eradication is possible.

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality) — disable PSD upload endpoints as unneeded functionality until patch is confirmed, CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Run 'pip-audit --requirement requirements.txt' or 'pip list | grep -i pillow' across every virtualenv, container image, and CI runner. For containerized workloads: 'docker exec pip show Pillow'. Block PSD uploads at the application layer by adding a file-extension and MIME-type denylist in your web framework (e.g., Flask/Django request validation) or at the reverse proxy with an nginx 'if ($content_type ~* "image/vnd.adobe.photoshop")' deny rule — no SIEM required.

Evidence: Before disabling upload endpoints, capture: (1) web server access logs (Apache/nginx) filtered for requests with Content-Type 'image/vnd.adobe.photoshop' or file extensions '.psd'; (2) application-level upload logs showing filenames, sizes, source IPs, and authenticated user/session IDs for all PSD files processed since the vulnerability window opened; (3) a snapshot of currently installed Pillow version per environment via 'pip show Pillow > pillow_inventory_$(hostname)_$(date +%F).txt' to establish baseline for later comparison.

Detection — Search application logs for PSD file processing activity, particularly large or malformed files. Review crash logs, segmentation fault signals, or unexpected process restarts in services that invoke Pillow image parsing. No specific CVE-linked IOC hashes are available at this time.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate application crash signals and anomalous file processing events to determine whether CVE-2026-42311 has been actively triggered.

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-3 (Content of Audit Records), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Query web server access logs: 'grep -E "\.psd|vnd\.adobe\.photoshop" /var/log/nginx/access.log | awk "{print $1, $7, $9, $10}"' to extract source IP, URI, response code, and bytes sent. For crash detection: 'journalctl -u --since "2026-03-04" | grep -iE "segfault|signal 11|killed|oom|restart"' on systemd hosts. On Python process hosts, search for core dump files: 'find /var/crash /tmp /home -name "core.*" -newer /etc/pillow-baseline 2>/dev/null'. Use osquery: 'SELECT pid, name, cmdline, start_time FROM processes WHERE name LIKE "%python%";' to identify currently running Pillow-dependent services.

Evidence: Exploit of CVE-2026-42311 via a malformed PSD file with invalid tile extents (integer overflow in PSD parser) would produce: (1) SIGSEGV (signal 11) or SIGABRT crash entries in /var/log/syslog or journald for the Python worker process; (2) Linux kernel OOM or segfault messages: 'grep "segfault\|python" /var/log/kern.log'; (3) unusually large PSD files (crafted files typically exceed normal tile size bounds — flag uploads >10MB or with anomalous PSD chunk headers); (4) Python traceback logs referencing Pillow's PsdImagePlugin.py or ImagingDecoder internals if exception handling caught a partial crash; (5) web application error logs (e.g., Gunicorn/uWSGI stderr) showing HTTP 500 responses correlated with PSD upload requests.

Eradication — Monitor the Pillow PyPI page (https://pypi.org/project/Pillow/) and the official GitHub repository (https://github.com/python-pillow/Pillow) for a patched release addressing GHSA-pwv6-vv43-88gr. Apply the patched version via 'pip install --upgrade Pillow' once confirmed. Verify via 'pip show Pillow' post-upgrade.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the vulnerable Pillow version from all affected environments and confirm the integer overflow flaw (CVE-2026-42311) is eliminated by deploying the vendor-confirmed patched release.

Controls: NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-3 (Configuration Change Control), 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: Pin the patched Pillow version in requirements.txt and run 'pip install --upgrade Pillow=={patched_version} && pip show Pillow | grep Version' to confirm. Verify package integrity against the PyPI-published SHA256 hash: 'pip download Pillow=={patched_version} -d /tmp/pillow_verify && pip hash /tmp/pillow_verify/Pillow-*.whl' then cross-check against the hash listed on https://pypi.org/project/Pillow/#files. For air-gapped or locked environments, use 'pip-audit --fix' with a pinned advisory database. Run 'pip-audit' again post-upgrade and confirm GHSA-pwv6-vv43-88gr no longer appears in the output.

Evidence: Before upgrading, preserve: (1) exact output of 'pip show Pillow' from each affected environment (version, location, dependencies) as eradication baseline evidence; (2) hash of the vulnerable Pillow wheel or egg file currently installed: 'sha256sum $(pip show -f Pillow | grep Location | awk "{print $2}")/Pillow*'; (3) any malformed PSD files submitted to the application — preserve originals in write-protected forensic storage as they represent exploit artifacts specific to CVE-2026-42311's integer overflow trigger condition in PSD tile extent parsing.

Recovery — After upgrading, re-run dependency scans to confirm the patched version is active across all environments, including containers and CI/CD pipelines. Retest image processing workflows with known-good PSD files to confirm functional integrity. Monitor application error rates for 24-48 hours post-deployment.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore normal Pillow-dependent image processing operations after confirming the patched version is deployed across all execution environments and verifying no residual instability from prior exploitation.

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST CP-10 (System Recovery and Reconstitution), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: For container environments: rebuild all images from scratch with the pinned patched Pillow version and run 'docker inspect | grep -A5 Pillow' or re-run 'pip list' inside the rebuilt container to verify. For CI/CD: add a mandatory pipeline gate — 'pip-audit --requirement requirements.txt --fail-on-vuln' — that blocks deployment if any Pillow advisory reappears. Monitor Gunicorn/uWSGI error rates with 'tail -f /var/log/gunicorn/error.log | grep -c

"500\|exception"' per 15-minute window for the 24-48 hour watch period.

Evidence: During the recovery watch period, retain: (1) application HTTP error rate logs (5xx responses) segmented by endpoint — a spike in the PSD-processing endpoint post-patch could indicate incomplete remediation or a residual exploit attempt; (2) output of 'pip-audit' run in each environment post-upgrade confirming clean status — save as 'pip_audit_post_patch_$(hostname)_$(date +%F).txt'; (3) Python process crash logs (journald/syslog) for the 48-hour window to confirm SIGSEGV/SIGABRT signals cease after patching, distinguishing pre-patch exploitation noise from post-patch stability.

Post-Incident — Review image input validation controls: enforce file type allowlists, size limits, and consider sandboxed image processing for untrusted input. Evaluate whether software composition analysis (SCA) tooling is integrated into your CI/CD pipeline to catch future library-level CVEs before reaching production.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: translate lessons from CVE-2026-42311 into durable controls that prevent untrusted PSD files from reaching Pillow's parser and ensure future Pillow CVEs are detected before reaching production.

Controls: NIST SI-10 (Information Input Validation), NIST SI-2 (Flaw Remediation), NIST IR-4 (Incident Handling), 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), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Implement magic-byte validation at upload ingestion — do not trust file extension alone; PSD files begin with '38 42 50 53' (hex) — validate this header server-side using Python's 'imghdr' or a custom check before passing to Pillow. Enforce a hard upload size cap (e.g., 10MB) in your web framework to reduce the feasibility of crafting oversized tile extent payloads. For sandboxed processing, run Pillow image operations in a subprocess with 'resource.setrlimit(resource.RLIMIT_AS, ...)' memory limits or a Docker container with '--memory=256m --cpus=0.5 --no-new-privileges'. Integrate 'pip-audit' or OWASP Dependency-Check into your CI pipeline as a blocking step on every pull request touching requirements.txt.

Evidence: Lessons-learned documentation should include: (1) a timeline of when the vulnerable Pillow version was introduced (check git history: 'git log -p requirements.txt | grep -A2 -B2 Pillow') to calculate dwell time of the vulnerable dependency; (2) list of all PSD files processed during the vulnerability window, extracted from web server access logs and application upload records, to assess potential exploitation exposure; (3) results of the final 'pip-audit' clean run and patched version inventory across all environments as closure evidence for the incident record per NIST IR-5 (Incident Monitoring).

Detection Guidance

No public proof-of-concept or active exploitation indicators are confirmed at this time. Detection focus should be behavioral: monitor for application crashes, segmentation faults, or unexpected memory errors in services that invoke Pillow's PSD parsing code. In containerized environments, watch for container restarts correlated with image upload events. For Python applications, enable exception logging around image processing calls and alert on unhandled exceptions from Pillow's PsdImagePlugin module. Dependency scanning tools such as pip-audit, Snyk, or GitHub Dependabot will flag vulnerable Pillow versions once the advisory is indexed. GHSA-pwv6-vv43-88gr on OSV.dev is the authoritative tracking identifier.

Framework Mappings

MITRE-ATTACK

- **T1203** — Exploitation for Client Execution

NIST-800-53R5

- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-16** — Memory Protection
- **SR-2** — Supply Chain Risk Management Plan

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **15.1** — Establish and Maintain an Inventory of Service Providers

ISO-27001-2022

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

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1203	Exploitation for Client Execution	Execution

Sources

Source	URL	Tier
osv	https://osv.dev/vulnerability/GHSA-pwv6-vv43-88gr	T3
CVE-2026-42311 Pillow PSD Tile Extents integer overflow	https://vuldb.com/vuln/361106	T3
CVE-2026-2311 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-2311	T1
Security Bulletin Archive - HP Support	https://support.hp.com/gb-en/security-bulletins?utm_term%2525252525...	T3

Source	URL	Tier
CVE-2021-42311: Microsoft Defender For IoT RCE Vulnerability	https://www.sentinelone.com/vulnerability-database/cve-2021-42311/	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-42311	T1

Generated 2026-05-06 09:04 UTC by TJS Security Command Center