contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

Pillow TIFF overflow

CVE-2026-112393 · GHSA-332p-7hrc-92wf

Assessment & Action Plan

A heap overflow in Python's most widely used imaging library: who is exposed, how to check, and the two sibling CVEs fixed in the same release.

PREPARED BY	Hung Le, NamiQ	DATE	22 October 2026
FACTS AS OF	22 October 2026 (CISA KEV 2026.10.21)	VERSION	1.0
SEVERITY	**Critical: CVSS 4.0 9.2**	STATUS	Fixed in Pillow 13.0.0

COMPANION ARTICLE namiq.io/insights/pillow-cve-2026-112393-tiff-heap-overflow

NamiQ © 2026 NamiQ. All rights reserved. This document is the property of NamiQ. It may be shared unaltered, with attribution.
namiq.io

Executive summary

In brief

CVE-2026-112393 is a heap buffer overflow in **Pillow**, the standard Python imaging library, in the code that writes TIFF files. It has been present since **Pillow 2.0.0 (2013)** and is fixed in **Pillow 13.0.0**, released on 16 October 2026.

- **It bites one specific pattern:** an application that turns an image someone else supplied into a **compressed TIFF**, written to an **in-memory or streamed target** (`io.BytesIO`, an HTTP response, a storage stream) rather than to a file on disk. Saving to a file path is not affected.
- **The attacker controls TIFF metadata, not pixels:** a crafted out-of-line tag (ImageDescription, DocumentName, XMP) or an embedded ICC colour profile. A second, independent defect leaks stale heap bytes into any compressed TIFF over 64 KiB written the same way.
- **Severity: 9.2 Critical (CVSS 4.0)**, network-reachable with no login, but with **high attack complexity** and **attack prerequisites present**. A crash reproducer is public; the reporters say they turned the bug into code execution with help from a separate memory leak, and did not publish that chain.
- **Not known to be exploited:** not in CISA's KEV catalog (version 2026.10.21), and we found no report of exploitation in the wild as of 22 October.
- **The same release fixes two more 9.2 CVEs.** One of them, **CVE-2026-112403** in JPEG `thumbnail()`, is on the code path almost every upload feature uses, and is likely the bigger exposure for most teams.

Recommendation: upgrade every Pillow that processes untrusted images to **13.0.0** this week, starting with anything that produces TIFFs or thumbnails from uploads. Pillow 13.0.0 **drops Python 3.10**, so check that first. Where you cannot upgrade yet, apply the code-level workarounds in section 8.3, which we tested.

At a glance

	CVE-2026-112393
Advisory	GHSA-332p-7hrc-92wf (Pillow maintainers, published 17 October 2026)
Component	`_tiffWriteProc` in `src/libImaging/TiffDecode.c`, the libtiff write callback used for non-file targets
Weakness	CWE-122 Heap-based Buffer Overflow, plus disclosure of uninitialised heap bytes
Affected	Pillow >= 2.0.0, < 13.0.0
Fixed in	**13.0.0** only. No 12.x release carries the fix
Severity	9.2 Critical, CVSS 4.0 (`AV:N/AC:H/AT:P/PR:N/UI:N/VC:H/VI:H/VA:H`)

 © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

	CVE-2026-112393
Exploitation	Crash reproducer public; code-execution chain reported, not published; none in the wild reported
CISA KEV	Not listed (catalog 2026.10.21)
Not affected	Saves to a file path or to an object with a real file descriptor; uncompressed TIFF output

1. What the vulnerability is

Pillow writes compressed TIFF files through **libtiff**. When the destination is a real file, Pillow hands libtiff the file descriptor and libtiff writes to it directly. When the destination has **no usable `fileno()`** (an `io.BytesIO`, a web framework's response stream, a cloud-storage upload stream), Pillow instead gives libtiff a **growable memory buffer** and a set of callbacks to write into it. `_tiffWriteProc` is the write callback.

1.1 The defect

libtiff does not always write sequentially. To lay out a file it **seeks forward past the current end** and writes there, for example when it places a tag value that is too large to fit inside the directory entry (an "out-of-line" value). The callback grew its buffer based on the buffer's **previous size** rather than on the **position being written to**, so a write after such a seek could land **past the end of the allocation**.

The advisory's account of the consequence: the bytes written out of bounds are the **tail of a tag value the attacker chose**, and the overflow is the size of the image file directory (12 bytes per entry the upload carries: 126 bytes with the minimal reproducer, 2,384 bytes with a 331-entry directory). The defect was introduced in **commit e782fe7 in 2013** and has been present since Pillow 2.0.0.

1.2 Two effects, with different preconditions

	Heap overflow	Stale-heap disclosure
What happens	Out-of-bounds write of attacker-chosen bytes past a ~128 KiB heap block	One byte of previously freed heap memory per alignment gap is copied into the output file
Needs a crafted file?	**Yes.** An attacker-controlled out-of-line tag value must reach libtiff	**No.** Any compressed TIFF over 64 KiB written to a non-file target

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

	Heap overflow	Stale-heap disclosure
Where the attacker's data comes from	ImageDescription, DocumentName or XMP, when the application re-saves the uploaded TIFF or copies its tags An **embedded ICC profile**, which survives `convert()`, `resize()`, `thumbnail()` and `copy()`	Not attacker-controlled: whatever other requests freed earlier
Worst case	Process crash (glibc detects it and aborts); code execution demonstrated when combined with an address leak from another bug	Low-bandwidth leak of other requests' data at predictable offsets in returned files

Why compression matters. Pillow writes an **uncompressed** TIFF with its own Python writer and only uses libtiff when the output is compressed. Compression is either requested by the application (`compression="tiff_lzw"` and similar) or **inherited from the source image**: an uploaded compressed TIFF re-saved as TIFF keeps its compression by default. We confirmed this in the Pillow source (`TiffImagePlugin._save`).

1.3 What is not affected

- **Saving to a file path**, or to a file object with a real descriptor (`open(...)`, `tempfile.TemporaryFile()`). These go through libtiff's own file I/O, not the vulnerable callback.
- **Reading TIFFs.** `Image.open()` and decoding use different code; this CVE is on the write path only. (Earlier decode-side TIFF overflows, such as CVE-2020-35654 and CVE-2021-25289, are long fixed.)
- **Output in other formats.** Producing JPEG, PNG or WebP thumbnails does not reach `_tiffWriteProc`. It may reach CVE-2026-112403 instead (section 4).
- **Uncompressed TIFF output**, because libtiff is not used to write it.

1.4 Weakness and technique mapping

ID	Name	How it applies here
CWE-122	Heap-based Buffer Overflow	Writes land past the end of the reallocated memory buffer. The CWE assigned in the advisory.
CWE-131	Incorrect Calculation of Buffer Size	The buffer is sized from its previous size, not the write position.
CWE-908	Use of Uninitialized Resource	Alignment gaps in the grown buffer are never written, so stale heap bytes reach the output.
ATT&CK T1190	Exploit Public-Facing Application	An upload endpoint that converts images is the entry point.

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

ID	Name	How it applies here
ATT&CK T1499.004	Application or System Exploitation	Even without code execution, a crafted upload kills the worker process.

CWE-122 is from the advisory. The other mappings are NamiQ's own.

2. Severity: what the scores mean

Different numbers for this CVE are already circulating. Here is each one and where it comes from:

Source	Rating	Note
GitHub Advisory Database (reviewed entry)	**9.2 Critical**, CVSS 4.0	The score the Pillow maintainers attached as CNA. Vector below.
Pillow's repository advisory page	"High"	The severity label on the maintainers' own advisory page. Same advisory, different label.
NVD	No NVD score yet	Status "Undergoing Analysis". The only score in the record is the CNA's 9.2.
CISA ADP (SSVC, 19 Oct)	Exploitation: **PoC**; automatable: **no**; technical impact: **total**	"PoC" reflects the public crash reproducer in the advisory.
EPSS (via GitHub, 22 Oct)	0.528%, 44th percentile	Modelled probability of exploitation activity in the next 30 days. A snapshot; it moves daily.
Some vulnerability feeds	9.5 Critical	We found **no primary source** for 9.5. Use 9.2.

The vector, decoded: `CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`

Metric	Value	What it means for you
Attack vector	Network	Reachable remotely, if the application takes images from the network.
Attack complexity	**High**	Turning the overflow into more than a crash needs heap grooming and, per the reporters, a separate address leak.
Attack requirements	**Present**	The application must convert to compressed TIFF, write to a non-file target, and pass the attacker's tag or ICC profile through.
Privileges / user interaction	None / None	An anonymous upload is enough if the endpoint is public.
Impact on the vulnerable system	High / High / High	Code execution in the worker process, in the worst case.

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

Metric	Value	What it means for you
Impact on other systems	None	Scored for the process only. What the worker can reach afterwards is your question.

3. Real-world exploitability

3.1 Timeline

When	What happened	Source
2013	The defect is introduced (commit e782fe7) and ships in Pillow 2.0.0.	GHSA
16 Oct 2026	**Pillow 13.0.0** is released on PyPI with fixes for this and the two sibling CVEs.	PyPI; release notes
17 Oct	Pillow maintainers publish **GHSA-332p-7hrc-92wf**, with a public reproducer for the crash and the leak. Reporters credited: carlini and Tomer-PL.	GHSA
19 Oct	CVE-2026-112393 published in NVD ("Undergoing Analysis"). CISA's SSVC entry: exploitation **PoC**, not automatable, total technical impact.	NVD
21 Oct	The GitHub Advisory Database reviews the entry: **9.2 Critical**, CVSS 4.0. CISA KEV catalog 2026.10.21 does **not** list it.	GitHub; CISA KEV
22 Oct	OX Security publishes a summary of all three Pillow CVEs. Debian's tracker shows the fix in **sid only** (13.0.0-1); bookworm, trixie and forky are still listed as vulnerable.	OX Security; Debian

3.2 What an attacker needs

All of the following must be true for the overflow. Each one narrows the population of exposed applications:

Condition	How common it is
The application accepts images from someone else	Very common: avatars, document intake, scanning, ML inference.
It produces a TIFF from them	Uncommon on the web, common in **document imaging, OCR, fax, archiving, GIS and medical or scientific pipelines**.
The TIFF is compressed	Usually, where TIFF is used at all. Inherited automatically when re-saving an uploaded compressed TIFF.
It is written to a non-file target	Common: returning the file in an HTTP response or uploading it to object storage without touching disk is the idiomatic pattern.

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

Condition	How common it is
An attacker-controlled out-of-line tag or ICC profile reaches the save	Happens by default when re-saving an uploaded TIFF, or when the source image carries an ICC profile.

The **stale-byte leak** needs only the second, third and fourth conditions, plus an output over 64 KiB. It is low-bandwidth, but it hands other requests' freed data to whoever receives the file. That matters most in **multi-tenant** services.

3.3 Exposure by application pattern

Pattern	Exposure	Why
Upload → convert to compressed TIFF → return or store from memory	**High**	Every condition met by default. The pattern in the advisory's own proof of concept.
Document or OCR pipeline normalising inputs to TIFF in memory	**High**	Same pattern, often fed from email or scanning, so the input is external.
Upload → JPEG/PNG/WebP thumbnail	Not affected by **this** CVE	But the JPEG `thumbnail()` path is affected by **CVE-2026-112403** (section 4). Upgrade anyway.
Converts to TIFF but writes to a file path or temp file	Not affected	Real file descriptors use libtiff's file I/O.
Converts to uncompressed TIFF	Not affected	Written by Pillow's Python writer, not libtiff.
Pillow present only as a dependency (e.g. matplotlib output)	Low	No path from untrusted input to a TIFF save. Patch in the normal cycle.

3.4 NamiQ assessment

> **Our reading, 22 October 2026**
>
> This is a **real memory-corruption bug with a public crash reproducer** in a library that is in nearly every Python estate. The code-execution claim is credible but **conditional**: the reporters needed an address leak from a separate bug, and they did not publish that chain. Nothing suggests exploitation in the wild.
>
> For **most web applications the exposure is narrow**, because they do not produce TIFFs. For **document-imaging, OCR, archiving and scientific pipelines the exposure is direct**, and those are often the systems that process files from outside the organisation by design.
>
> The practical answer is the same either way: **move to Pillow 13.0.0**, because the same release fixes a JPEG thumbnail overflow that does sit on the common path.

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

4. The same release fixes two more critical CVEs

Pillow 13.0.0 is the only fix for all three, and they should be planned as one change. Each is rated 9.2 Critical under CVSS 4.0.

	CVE-2026-112393	CVE-2026-112403	CVE-2026-112900
Advisory	GHSA-332p-7hrc-92wf	GHSA-8g5g-8vmr-9782	GHSA-w889-685w-c8fh
Where	TIFF **write** to a non-file target	JPEG **decode** via `draft()` / `thumbnail()`	EPS rendering through **Ghostscript**
Trigger	Crafted tag or ICC profile, compressed TIFF output	A standards-compliant 8-bit **lossless (SOF3) JPEG**, thumbnailed	An **attacker-influenced filename** passed to `Image.open()`
Extra condition	Non-file output target	Pillow linked to **libjpeg-turbo 3.x**, which the **official wheels bundle**	Ghostscript installed; full RCE needs control of a referenced argument file
Affected	2.0.0 to < 13.0.0	Up to and including 12.3.0	2.6.0 to < 13.0.0
Interim workaround	Write to a temp file, or uncompressed (8.3)	Call `im.load()` before `thumbnail()`, or pass `reducing_gap=None`	Never derive the filename from user input; remove Ghostscript where not needed

Why CVE-2026-112403 probably matters more to you. `Image.thumbnail()` is how nearly every upload feature makes a preview, and it calls `draft()` by default. Opening the file with `formats=["JPEG"]` does not help, because the malicious file is a valid JPEG. Distribution builds of Pillow that link **libjpeg-turbo 2.x** or IJG libjpeg reject the file before any row is decoded and are not affected; the official PyPI wheels bundle 3.x (our check: Pillow 12.3.0's wheel reports libjpeg-turbo 3.1.4.1).

CVE-2026-112900 is narrower: it needs an attacker to influence the **name** of the file Pillow opens, which happens when an upload handler stores files under the name the user supplied. A name beginning with `-` or `@` is read by Ghostscript as an option or an argument file, even with `-dSAFER`.

5. Should you patch now?

Yes, this week for anything that handles untrusted images; in the normal cycle for the rest. The decision per system:

Your situation	Recommended action	Target
Converts external images to compressed TIFF, writing to memory or a stream	Upgrade to 13.0.0. Until then, apply the temp-file workaround (8.3).	Within 72 h

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

Your situation	Recommended action	Target
Makes thumbnails or previews from uploaded JPEGs, using PyPI wheels	Upgrade to 13.0.0 (CVE-2026-112403). Until then, `im.load()` before `thumbnail()`.	Within 72 h
Any other service that processes images from outside the organisation	Upgrade to 13.0.0.	Within 7 days
Pinned to Python 3.10 (Pillow 13 does not support it)	Upgrade Python, or use your distribution's patched package when one ships, and apply the workarounds meanwhile.	Plan this week
Pillow only as a transitive dependency, no untrusted images	Upgrade in the normal release cycle.	Within 30 days
Vendor products embedding Python and Pillow	Ask the vendor; do not patch inside an appliance yourself.	As vendors publish

Targets are NamiQ's recommendation for a typical small or mid-size estate, not a regulatory deadline. A KEV listing would set formal deadlines for US federal agencies.

6. Exposure assessment

6.1 Inventory: where Pillow hides

- **Direct use:** upload handlers, thumbnailers, document and OCR pipelines, report generators.
- **Frameworks and applications that require it:** Django's `ImageField`, Wagtail, Odoo, Plone, and most CMSs with image handling.
- **Data and ML stacks:** matplotlib and torchvision depend on Pillow; so do many notebook, dashboard and inference-serving images.
- **Containers and serverless bundles:** the version is whatever was resolved at build time, often months ago. Check the running image, not `requirements.txt`.
- **Operating-system packages:** `python3-pil` / `python3-pillow` from your distribution, used by system tools and by applications that run on the system Python.
- **Vendored copies:** desktop apps, appliances and tools shipping their own Python runtime.

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

6.2 Verify the version you actually run

Check the interpreter that runs the service, not the one on your path. Distribution packages may backport fixes without changing the version string, so for `python3-pil` read the security tracker or changelog. As of 22 October, **Debian's tracker lists bookworm (9.4.0), trixie (11.1.0) and forky (12.3.0) as vulnerable** and sid (13.0.0) as fixed.

```
# 1. Version and bundled libraries, from the service's own interpreter
python -c "import PIL; from PIL import features; print(PIL.__version__, 'libjpeg-turbo',
features.version('libjpeg_turbo'), 'libtiff', features.version('libtiff'))"

# 2. Full report: where Pillow is loaded from and what it was built with
python -m PIL --report

# 3. Lockfiles and requirements across a repository
grep -rni pillow --include='requirements*.txt' --include='poetry.lock' --include='uv.lock' --
include='Pipfile.lock' --include='pyproject.toml' .

# 4. Containers: the running image, not the Dockerfile
docker exec <container> python -c "import PIL; print(PIL.__version__)"

# 5. Debian / Ubuntu system package
dpkg -l | grep -E 'python3-pil'
```

Any version below **13.0.0** from PyPI is affected by all three CVEs. Our check of the commands above: on Pillow 12.3.0 from PyPI, command 1 prints `12.3.0 libjpeg-turbo 3.1.4.1 libtiff 4.7.1`.

6.3 Find the vulnerable code patterns

A version below 13.0.0 tells you the library is vulnerable; these searches tell you whether your code reaches it. Review every hit whose input can come from outside:

```
# CVE-2026-112393: TIFF output
grep -rnE --include='*.py' "save\(.*['\"](TIFF|tiff)['\"]|\.tiff?['\"]" .

# CVE-2026-112403: thumbnail() / draft() on JPEG input
grep -rnE --include='*.py' "\.thumbnail\(|\.draft\(" .

# CVE-2026-112900: EPS through Ghostscript, and whether Ghostscript is installed
grep -rnE --include='*.py' "EpsImagePlugin|\.eps['\"]" .
command -v gs
```

For each TIFF hit, ask: is the target a `BytesIO`, a response object or a storage stream? Is the output compressed, or is the input an uploaded TIFF? For each `thumbnail()` hit: can the input be a JPEG from outside? Searches miss indirect calls (helpers, third-party libraries), so treat them as a starting point, not proof of absence.

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

7. How early summaries got this wrong

Automated and early write-ups of this CVE, including ones we reviewed while preparing this note, repeated several errors. Each one would change a decision:

Claim seen in early summaries	What the primary sources say
CVSS 9.5	9.2 (CVSS 4.0), from the CNA. No primary source for 9.5.
"NVD rates it low"	NVD has not scored it. Its record shows only the CNA's 9.2.
The attacker controls pixel data	The overflow comes from **tag values or an ICC profile**. Pixels only shape the layout.
Requires the application to opt into compression	Compression is **inherited** from an uploaded compressed TIFF by default.
Only `BytesIO` is affected	**Any target without a usable `fileno()`**: HTTP responses, storage streams, other file-like objects.
No public proof of concept	A **crash and leak reproducer is public** in the advisory. The code-execution chain is not.
Fixed in 13.0.0 "or the next patch release"	**13.0.0 only.** No 12.x backport, and 13.0.0 drops Python 3.10.
Read-only applications are safe	Safe from this CVE, but `thumbnail()` on JPEG input is exposed to **CVE-2026-112403**.
Related ICS/OT heap overflows and integer-overflow attack patterns	Graph neighbours by weakness class, not related to this bug. Ignore for this decision.

8. Action plan

8.1 By urgency

Immediate (today)

- Run the version check (6.2) and the code search (6.3) on every service that **accepts images from outside**.
- For each hit, decide: upgrade now, or apply the workaround from 8.3 now and upgrade within the week. Give each decision an owner.
- Check for Python 3.10 pins: they block the upgrade and need a plan of their own.

Short term (within 72 hours)

- Upgrade to **Pillow 13.0.0** every service that converts untrusted images to TIFF or thumbnails untrusted JPEGs.
- Rebuild container images and serverless bundles; **verify the running image** (6.2, step 4).
- Run your test suite: 13.0.0 removes some deprecated APIs and changes a few behaviours (for example, XMP info is no longer a tuple).

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

Medium term (within 7–30 days)

- Upgrade every other environment, including CI images, notebooks and internal tools.
- Track distribution packages (`python3-pil`) until your release ships a fix, and apply it.
- Add a dependency check that fails builds below the fixed baseline: `pip-audit`, Dependabot, Renovate or your SCA tool.
- Record all three CVEs in your vulnerability register, with an owner and evidence of closure per service.

8.2 Upgrade procedure (per service)

1. Confirm the service runs on **Python 3.11 or later**; Pillow 13.0.0 does not support 3.10.
2. Raise the pin: `pip install --upgrade "pillow>=13.0.0"` (or the equivalent lockfile update), and rebuild.
3. Run the test suite, paying attention to image I/O, EXIF/XMP handling and any use of removed APIs.
4. Deploy, then verify from the running service: version 13.0.0 or later (6.2).
5. Remove interim workarounds only after verification, and record the evidence.

8.3 If you cannot upgrade yet: tested workarounds

These remove the vulnerable path in your own code; they do not fix the library. We tested the first two against Pillow 12.3.0 (details below the table).

Workaround	Closes	Cost and caveats
Write TIFFs to a real temporary file, then read the bytes back (code below)	112393: overflow and leak	A disk write per conversion. The temp file must be a real descriptor: `tempfile.TemporaryFile()`, not `SpooledTemporaryFile`.
Save TIFFs uncompressed (`compression="raw"`)	112393: overflow and leak	Much larger files. Must be explicit: compression is otherwise inherited.
Strip metadata before saving: drop `icc_profile` from `im.info`; do not copy uploaded tags	112393: overflow only	Removes colour management. Easy to miss a path; prefer the temp file.
Call `im.load()` before `thumbnail()`, or pass `reducing_gap=None`	112403	Decodes at full size: slower and more memory per image. Keep `Image.MAX_IMAGE_PIXELS` in place.
Store uploads under server-generated names; open with `formats=[...]` excluding EPS; remove Ghostscript where not needed	112900	None for most applications.

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

Workaround	Closes	Cost and caveats
Run image processing in a separate, low-privilege worker with no secrets and no outbound network	Limits impact of all three	Architecture change; worth having for any image pipeline regardless.

```
import tempfile

def tiff_bytes(im, **save_options):
    # A real file descriptor makes libtiff write through its own file I/O,
    # not the vulnerable in-memory callback (CVE-2026-112393).
    with tempfile.TemporaryFile() as f:
        im.save(f, "TIFF", **save_options)
        f.seek(0)
        return f.read()
```

Our test. Using the advisory's harmless leak scenario (a 300×300 noise image saved as LZW TIFF with odd-length text tags), we compared output saved to a file path with output saved to `BytesIO`, to `TemporaryFile()` and as uncompressed TIFF. On Pillow 12.3.0, the `BytesIO` output contained stale heap bytes in **3 of 20 runs**; the temporary-file and uncompressed variants were identical to the file-path output in **all 20**. On 13.0.0, `BytesIO` was identical in all 20. The leak is intermittent because stale memory is often zero, so **a passing leak test proves nothing**. Rely on the version check.

8.4 Detection and hunting

Where	What to look for
Application and process logs	Image-processing workers dying with **SIGABRT** (signal 6), or glibc messages such as `malloc(): corrupted top size`, `realloc(): invalid next size` or `double free or corruption`. Core dumps from Python processes in image pipelines (`coredumpctl list` on systemd hosts).
Uploaded files	TIFF uploads with **hundreds of directory entries**, or **very large text tags** (ImageDescription, DocumentName, XMP). The reporters' code-execution file was a 4.7 MB TIFF with a 63,942-byte XMP packet and about 500 private entries. JPEGs using **lossless SOF3** encoding (marker `0xFFC3`), which are rare in normal uploads (CVE-2026-112403).
Upload filenames	Names beginning with `-` or `@` (CVE-2026-112900).
Host / EDR	Image workers starting shells, downloading tools or making outbound connections.

A crash is the most likely visible sign of an attempt, because the minimal reproducer aborts the process. Treat a cluster of worker aborts on an image endpoint as a security event, not just an availability one. **If you find signs of compromise:** isolate the worker, preserve logs, uploads and core dumps, rotate every credential the process could read, and rebuild from a known-good image.

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

8.5 Definition of done

- Every service that processes untrusted images runs **Pillow 13.0.0 or later**, verified from the running service, with evidence recorded.
- Every Python 3.10 environment has a dated plan, and the workarounds from 8.3 are in place until it is upgraded.
- Distribution-packaged Pillow is tracked until the vendor ships a fix.
- A build-time dependency check fails anything below the fixed baseline.
- Worker crash logs from image endpoints since **17 October** have been reviewed.

9. What we do not know yet

- **Whether the code-execution chain becomes public**, or someone finds a way to do it without a second bug. Either would raise the urgency.
- **NVD's own score.** The record was still "Undergoing Analysis" on 22 October.
- **Distribution fixes.** Debian stable releases were still listed as vulnerable on 22 October; other distributions' status should be checked at the source.
- **Products embedding Pillow.** Vendor statements are still to come.

10. References

[1] Pillow — GHSA-332p-7hrc-92wf, CVE-2026-112393: heap buffer overflow in _tiffWriteProc
https://github.com/python-pillow/Pillow/security/advisories/GHSA-332p-7hrc-92wf

[2] GitHub Advisory Database — GHSA-332p-7hrc-92wf (reviewed entry, CVSS 4.0 9.2, EPSS)
https://github.com/advisories/GHSA-332p-7hrc-92wf

[3] NVD — CVE-2026-112393
https://nvd.nist.gov/vuln/detail/CVE-2026-112393

[4] Pillow — GHSA-8g5g-8vmr-9782, CVE-2026-112403: heap buffer overflow in ImagingJpegDecode
https://github.com/python-pillow/Pillow/security/advisories/GHSA-8g5g-8vmr-9782

[5] Pillow — GHSA-w889-685w-c8fh, CVE-2026-112900: argument injection in the Ghostscript call
https://github.com/python-pillow/Pillow/security/advisories/GHSA-w889-685w-c8fh

[6] Pillow 13.0.0 release notes
https://github.com/python-pillow/Pillow/releases/tag/13.0.0

[7] Pillow fix commit 11582e5
https://github.com/python-pillow/Pillow/commit/11582e5ffbdceb973d5ee593f3105af5a9852b40

[8] Pillow on PyPI
https://pypi.org/project/pillow/

[9] CISA — Known Exploited Vulnerabilities catalog
https://www.cisa.gov/known-exploited-vulnerabilities-catalog

[10] Debian security tracker — CVE-2026-112393
https://security-tracker.debian.org/tracker/CVE-2026-112393

NamiQ © 2026 NamiQ. All rights reserved. Property of NamiQ. May be shared unaltered with attribution.

[11] OX Security — Three critical Pillow CVEs (22 Oct 2026)
https://www.ox.security/blog/three-critical-pillow-cves-rce-and-heap-overflows-in-pythons-imaging-library-cve-2026-112900-cve-2026-112393-cve-2026-112403/

[12] NamiQ — Pillow CVE-2026-112393 (companion article)
https://namiq.io/insights/pillow-cve-2026-112393-tiff-heap-overflow

About this document

Prepared by **Hung Le**, NamiQ. NamiQ provides managed IT and security for growing businesses. This assessment reflects public information as of **22 October 2026**; check the references before acting on anything time-sensitive.

© 2026 NamiQ. This document is the property of NamiQ and may be shared unaltered, with attribution. It is not legal advice and does not guarantee the security of any system.

namiq.io · namiq.io/contact

Check your own servers and projects

The facts above are as of 22 October 2026. To compare your own estate with this CVE, the open-source lattice-scan command line checks the parts it can see. Naming the CVEs needs no key, only the CVE ids leave your machine, and a match is a candidate for a person to confirm, not a finding.

On a server: run it on the server itself, not on the laptop or CI runner that deploys to it. It reads installed packages (dpkg, rpm or apk), snaps, the names of running container images and the kernel version, compares them on the machine, and keeps the inventory there. A distribution can fix a CVE under an unchanged version number, so on Debian-family hosts read the changelog line the report shows.

```
lattice-scan --cve CVE-2026-112393
```

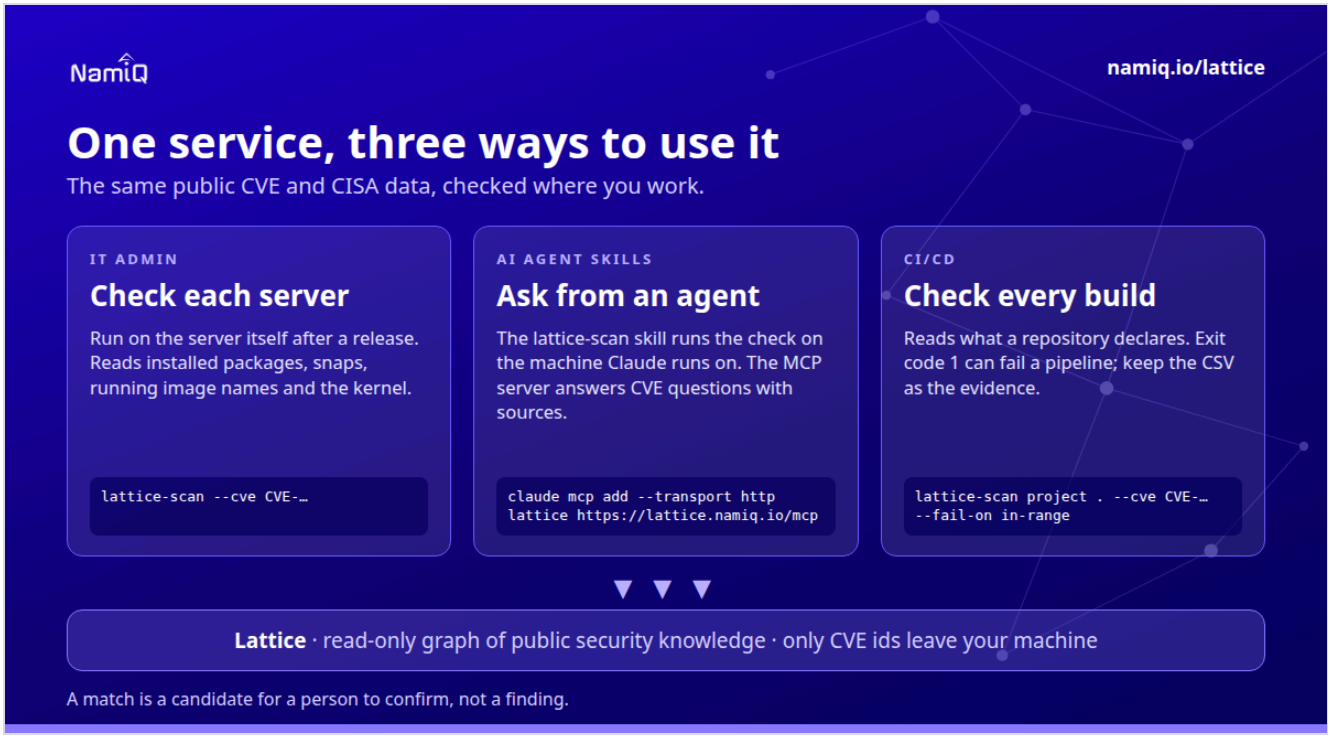
On a repository: the project check reads what it declares, in Python, JavaScript, Java, Rust, C and C++, Make, and Docker and compose files. It installs and runs nothing. A declared range or a tag without a patch number cannot show whether a fix is in, so such a row is undetermined and needs a person.

```
lattice-scan project . --cve CVE-2026-112393
```

A Pillow bundled inside a vendor product or a frozen application shows in neither check. The host check sees Pillow only where a distribution package installed it (for example python3-pil); a Pillow installed with pip is found by the project check on the requirements file that names it.

Who can use it, and how

For	What to do	Command
IT admin	Run the host check on each server after a release, and again when a new entry lands.	lattice-scan --cve CVE-2026-112393
AI agent skills	The lattice-scan skill runs the check on the machine Claude runs on. The Lattice MCP server answers questions about the CVE with its sources.	claude mcp add --transport http lattice https://lattice.namiq.io/mcp
CI/CD	Check the repository on every pull request. Exit code 1 fails the pipeline; keep the CSV as the evidence.	lattice-scan project . --cve CVE-2026-112393 --fail-on in-range



Where the same check is used. The three ways read the same public data; only CVE ids leave your machine.

Guide: namiq.io/lattice/guides/choose-your-channel. MCP server (io.namiq/lattice in the MCP registry): namiq.io/lattice. Package: pypi.org/project/lattice-scan, or namiq.io/apt.

Sources

- lattice-scan, the command line: https://github.com/namiq-ai/lattice-scan
- Which way into Lattice fits your job: https://namiq.io/lattice/guides/choose-your-channel
- Lattice, the MCP server: https://namiq.io/lattice