contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

WordPress Core exploited

CVE-2026-119977 · on CISA's exploited list since 3 November 2026

Analysis Report

WordPress 4.7 through 7.1.3 must be updated to 7.1.4 by 6 November 2026; if you do not run WordPress, stop here. Go to Dashboard → Updates on each site and apply the update now.

PREPARED BY	NamiQ	DATE	3 November 2026
FACTS AS OF	3 November 2026	VERSION	1.0
SEVERITY	**High: CVSS 3.1 8.1**	STATUS	On CISA's exploited list; due 6 November 2026

NamiQ © 2026 NamiQ. All rights reserved. This document is the property of NamiQ. It may be shared unaltered, with attribution.
Companion article: namiq.io/insights/wordpress-core-cve-2026-119977

The bottom line

> *WordPress 4.7 through 7.1.3 must be updated to 7.1.4 by 6 November 2026; if you do not run WordPress, stop here. Go to Dashboard → Updates on each site and apply the update now.*

At a glance

Fact	Detail
Product	WordPress Core (CVE-2026-119977)
Severity	High: 8.1 (CVSS 3.1)
CISA	On its exploited list since 3 November 2026. Deadline for US federal agencies: 6 November 2026.
Chance of exploitation (EPSS)	0.01, the estimated chance in the next 30 days, as at 3 November 2026
Ransomware use	Not recorded by CISA. That is not the same as none.
Fixed in	WordPress 7.1.4
Who looks first	Anyone running self-hosted WordPress on a version between 4.7 and 7.1.3. Sites where a hosting provider controls and applies core updates should ask the provider to confirm that 7.1.4 is applied.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run any WordPress site?	No action needed for this CVE.	Find out which version it runs (next question).
Is any WordPress installation running version 4.7 through 7.1.3?	Nothing to do if all installations are already on 7.1.4 or later.	Update to 7.1.4 now; the CISA deadline is 6 November 2026.
Can untrusted users reach the site over the internet?	Risk is reduced but the fix is still needed; update to 7.1.4 this week.	Every upgrade run while on an affected version is a window; update immediately.

What it is

During a WordPress core upgrade, temporary files are written to paths that are both predictable and reachable from the web. An unauthenticated attacker who times a request to that location during the upgrade window can place or run code on the server.

Why it matters

The CVSS score of 8.1 reflects that no login is needed: the record says any unauthenticated request timed to the upgrade window is sufficient. The staging path is described as predictable, so an attacker need not guess it. Our reading is that any publicly reachable WordPress installation that runs a core upgrade while on an affected version is exposed for the duration of that upgrade. The CISA deadline of 6 November 2026 is three days from the date it was added to the catalogue, which is unusually short.

What the vendor's advisory says

We read one page on wordpress.org. In short:

- The 7.1.4 release also fixes stored XSS in block markup, Customizer selective refresh, the Categories block, User Logins, Playlist and image blocks, and comments-pagination blocks.
- Also fixed: multiple stored XSS issues and a stored denial-of-service in the Interactivity API; stored XSS via Privacy Data Removal Requests; unpublished customizer changesets discoverable by unauthenticated users.
- Notes readable by privileged users who lack access to the post the note is attached to are also fixed. The release updates bundled root certificates due to the eMudhra revocation.
- The advisory does not give CVE identifiers or CVSS scores for the other issues in this release.
- Security fixes were backported to all branches currently eligible, back to 4.7. The advisory notes that only the most recent version of WordPress is actively supported.

Things people will get wrong

- The race-condition window is too short to exploit reliably. The record says the staging path is predictable and web-accessible, so an attacker can target a known location rather than guessing one. The CVSS 8.1 score reflects that no login is required.
- Automatic background updates mean we are already protected. Automatic updates apply the fix by upgrading to 7.1.4, but the upgrade run itself is the vulnerable window. Protection arrives when the upgrade to 7.1.4 completes successfully, not before it starts.
- Only sites that ran an upgrade recently are at risk. The record says the flaw is triggered by the upgrade process and affects versions 4.7 through 7.1.3. Any future upgrade run on an affected version opens the window again.

How we would triage it

This is our judgement.

Step 1: Find the WordPress version on each site

- In the WordPress dashboard, go to Dashboard → About to see the version number, or look for the version string in wp-includes/version.php on the server.
- List every WordPress installation you manage and note which are on 4.7 through 7.1.3; those are in scope.

Step 2: Update to WordPress 7.1.4

- In the WordPress dashboard, go to Dashboard → Updates and click Update Now. The vendor's advisory recommends this path.
- If the update cannot be applied from the dashboard, download the 7.1.4 package from WordPress.org and follow the manual upgrade steps, keeping wp-content and wp-config.php in place.

Step 3: Confirm the update succeeded

- After updating, check Dashboard → About or recheck wp-includes/version.php to confirm the version reads 7.1.4.
- If your hosting provider manages updates, ask them in writing to confirm that 7.1.4 is applied and to provide the version string as evidence before 6 November 2026.

Step 4: Check for signs of tampering if the site upgraded while exposed

- If the site ran a core upgrade while on 4.7–7.1.3 and was publicly reachable, compare core file checksums against the official WordPress package for that version.
- Look for unfamiliar PHP files in the site root or wp-content, and check Dashboard → Users for any administrator accounts you did not create.

To close it, read the version from the product itself, not from an old inventory, and save the output with the date. If you cannot update today, write down who owns it, the new date, and what limits exposure until then. CISA's wording adds that you should stop using the product if no mitigation is available.

Send this to your IT provider

Copy it, change the names and dates, and send it. It is a draft for you to check, not a template tested with any provider.

> *We are writing regarding CVE-2026-119977, a remote code execution vulnerability in WordPress Core affecting versions 4.7 through 7.1.3. CISA has listed it with a remediation deadline of 6 November 2026. The fix is WordPress 7.1.4. We ask you to confirm whether we run any WordPress installations, which versions they are on, and whether each has been updated to 7.1.4. Please provide written confirmation of the version running on each site and the date the update was or will be applied, by 6 November 2026.*

Signs it was used

An update closes the hole. It does not tell you whether someone used it first. CISA's required action points to its forensic triage requirements (BOD 26-04). Look for:

- Unfamiliar PHP files in the WordPress root or wp-content directories, especially any placed around the time of a recent core upgrade.
- New administrator-level user accounts in Dashboard → Users that your team did not create, particularly those appearing during or after a recent upgrade.
- Server access logs showing requests to temporary or staging paths in the WordPress root during any recent upgrade window.

The weakness behind it

The record classes this as CWE-367 (Time-of-check Time-of-use (TOCTOU) Race Condition).

The sources, in their words

CISA's catalogue entry:

> *WordPress WordPress Core contains a time-of-check time-of-use (TOCTOU) race condition vulnerability that could allow unauthenticated attackers to execute code on the server during the upgrade window. This is due to predictable, web-accessible staging paths used during full-package core upgrades.*

The CVE record, published by Wordfence on 31 October 2026:

> *WordPress Core is vulnerable to Remote Code Execution in versions 4.7 through 7.1.3. This is due to predictable, web-accessible staging paths used during full-package core upgrades. This makes it possible for unauthenticated attackers to execute code on the server during the upgrade window.*

The record lists 25 affected versions or ranges of WordPress Foundation WordPress. The first 8 are below; the rest are in the CVE record, so check your version against the whole list.

- 4.7.0 through 4.7.38
- 4.8.0 through 4.8.33
- 4.9.0 through 4.9.34
- 5.0.0 through 5.0.30
- 5.1.0 through 5.1.27
- 5.2.0 through 5.2.29
- 5.3.0 through 5.3.26
- 5.4.0 through 5.4.24

CISA's required action:

> *Apply mitigations in accordance with vendor instructions.*

Check your own project

If you build software, you can compare what a project declares with this CVE from a terminal. The tool reads requirements files, package.json, Maven, Gradle, Cargo, CMake and Dockerfile files, runs nothing, and only the CVE id leaves your machine. Naming the CVEs needs no key.

```
lattice-scan project . --cve CVE-2026-119977
```

Each row says in-range, outside-range, undetermined or lead, with the file and line. That is evidence about what a manifest declares, not about what runs, and a person decides. The verdicts, the CSV columns and a pipeline recipe are in the guide.

Check the servers you run

The same id can be checked against a host. Run this on the machine itself, not on the laptop 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 spot, and only the CVE ids leave the machine; the inventory stays on it, in ~/.lattice-scan. Naming the CVEs needs no key. Do it on each server after a deployment, and again when a new entry lands.

```
lattice-scan --cve CVE-2026-119977
```

A match is a candidate for a person to confirm, not a finding. A distribution can fix a CVE under an unchanged version number, so a match is a candidate; on Debian-family hosts the report also shows whether the package's own changelog names the CVE. It reads the names of the running images, not what is inside them; for what an image declares, use the project check on its Dockerfile.

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.

What we do not know yet

- The sources do not say how long a typical WordPress core upgrade window lasts, which affects the practical size of the exposure.
- The sources do not say which managed hosting providers have already applied 7.1.4 on behalf of their customers.
- The advisory lists many other issues fixed in 7.1.4 but does not provide CVE identifiers or severity scores for any of them.
- We have not tested this against any environment. Versions and scores are as the sources state them and as at 3 November 2026; the vendor's advisory wins where it differs.
- The sections headed "The bottom line", "Does this apply to you?", "What it is", "Why it matters", "Exposure by situation", "Things people will get wrong", "How we would triage it" and "Send this to your IT provider" are our interpretation of the sources, not a statement by CISA or the vendor.

Appendix 1. Severity decoded

CVE-2026-119977: score 8.1, `CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H` As the CVE record gives it. The record's own score can differ from the base score on the NVD record.

Metric	Value	What it means here
Attack vector	Network	Reachable over a network, with no need to be on the same segment.
Attack complexity	High	Needs conditions outside the attacker's control.
Privileges required	None	No account is needed.
User interaction	None	No one has to do anything.
Scope	Unchanged	Damage stays inside the vulnerable component.
Confidentiality	High	Total loss of confidentiality.
Integrity	High	Total loss of integrity.
Availability	High	Total loss of availability.

Appendix 2. Timeline

Date	Event
31 October 2026	CVE record published by Wordfence.
3 November 2026	CISA added it to its list of exploited vulnerabilities.
6 November 2026	CISA's deadline for US federal agencies.

Appendix 3. Exposure by situation

These priorities are our judgement from the attacker prerequisites in the sources. They are not vendor ratings.

Your situation	Priority	Timeframe
Self-hosted WordPress 4.7–7.1.3, site is internet-facing	High	Before 6 November 2026 — update to 7.1.4 immediately
Self-hosted WordPress 4.7–7.1.3, site is on a private or internal network only	Medium	This week — exposure is reduced but the fix is still required
Managed WordPress hosting where the provider controls and applies core updates	Medium	Confirm with provider that 7.1.4 is applied by 6 November 2026

Fully managed platform where the hosting service owns and applies all core updates	Low	Verify with provider; no direct action if 7.1.4 is confirmed
Self-hosted WordPress already on 7.1.4 or a later version	None	No action needed for this CVE

Appendix 4. Definition of done

- Every WordPress Core you run is listed, with its version.
- Each is outside the affected list, or the vendor's mitigation is in place.
- The version output or a screenshot is saved with the date.
- Your projects' declared dependencies were checked against this CVE, and every row has a written reason.
- Logs for the weeks before the update were searched, and anything odd was written down.
- Anything not done has an owner and a date.

Sources

- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- CVE record for CVE-2026-119977: https://www.cve.org/CVERecord?id=CVE-2026-119977
- FIRST: Exploit Prediction Scoring System: https://www.first.org/epss/
- Advisory or release notes read: wordpress.org/news/2026/10/wordpress-7-1-4/: https://wordpress.org/news/2026/10/wordpress-7-1-4/
- Reference in the CVE record: www.wordfence.com: https://www.wordfence.com/threat-intel/vulnerabilities/id/4b4da0cf-d9f4-47c4-a0e8-6c4dc5a6f250?source=cve
- Reference in the CVE record: core.trac.wordpress.org: https://core.trac.wordpress.org/browser/tags/7.1.3/src/wp-admin/includes/class-wp-upgrader.php#L378
- Reference in the CVE record: core.trac.wordpress.org: https://core.trac.wordpress.org/browser/tags/7.1.3/src/wp-admin/includes/class-core-upgrader.php#L162
- Reference in the CVE record: core.trac.wordpress.org: https://core.trac.wordpress.org/browser/tags/7.1.3/src/wp-admin/includes/update-core.php#L1549
- Reference in the CVE record: core.trac.wordpress.org: https://core.trac.wordpress.org/browser/tags/7.1.3/src/wp-admin/setup-config.php#L282