contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

addonsorg PDF Invoice for WooCommerce + Drag and Drop Template Builder: new CVE

CVE-2026-109220 · record published 31 October 2026

Analysis Report

PREPARED BY	NamiQ	DATE	5 November 2026
FACTS AS OF	5 November 2026	VERSION	1.0
SEVERITY	**Critical: CVSS 3.1 9.8**	STATUS	Not on CISA's exploited list; exploitation not recorded

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/addonsorg-pdf-invoice-for-woocommerce-drag-cve-2026-109220

Correction, 5 November 2026: an earlier version of this note described the last affected version as the first version that is not affected. The CVE record lists that version as affected. Read the vendor's page for the release that fixes it before you decide that you are safe.

The bottom line

> *Anyone running the PDF Invoice for WooCommerce + Drag and Drop Template Builder plugin must disable it now: no patched version is recorded. The score is 9.8 and no login is needed — any visitor to your store can trigger the flaw.*

At a glance

Fact	Detail
Product	addonsorg PDF Invoice for WooCommerce + Drag and Drop Template Builder (CVE-2026-109220)
Severity	Critical: 9.8 (CVSS 3.1)
Status	A new CVE record, published by Wordfence on 31 October 2026. Not on CISA's list of exploited vulnerabilities when we read it on 5 November 2026, which means nothing public says yet that anyone is using it.
Fixed in	No source we read names a fixed version
Who looks first	WordPress site owners running WooCommerce with this invoice plugin at version 7.1.1 or earlier. The record says a default install is enough to be at risk — no unusual configuration is required.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run WordPress with WooCommerce installed?	Nothing to do for this CVE.	Check whether this invoice plugin is installed (next question).
Is PDF Invoice for WooCommerce + Drag and Drop Template Builder installed and active?	Nothing to do for this CVE.	Check the plugin version (next question).

Is the plugin at version 7.1.1 or earlier?	No version above 7.1.1 is confirmed safe by any source; check with the vendor before reactivating.	Disable the plugin immediately — no patched version is recorded.
Is the WooCommerce Store REST API reachable by unauthenticated users?	Risk is reduced, but disable the plugin as a precaution regardless.	Any visitor to your site can reach the vulnerable code path; disable the plugin immediately.

What it is

An unauthenticated attacker submits an anonymous WooCommerce checkout with a crafted customer note. The plugin's invoice generator decodes that note and passes it to PHP's unserialize() without validation, allowing the attacker to run arbitrary code on the server.

Why it matters

The record scores this 9.8 (CVSS 3.1). The attack path begins at the public WooCommerce Store API, which is on by default. An anonymous buyer submits a cheque or direct bank transfer order with a crafted customer note; the plugin decodes and feeds that note to PHP's unserialize() without checking it (CWE-502). Our reading is that this is one of the most accessible paths to full server compromise in a WordPress context: no account, no prior access, and no unusual site configuration are needed. No vendor advisory was available when the record was read, so no workaround or confirmed fixed version exists.

Things people will get wrong

- Only logged-in customers can reach checkout, so the risk is limited The record says the attack uses the public WooCommerce Store API with an anonymous checkout; no account is needed to reach the vulnerable code path.
- The flaw is not urgent because it is not on CISA's exploited list A score of 9.8 with no login barrier warrants action before exploitation is confirmed. Absence from CISA's list reflects the timing of the read, not whether the flaw is safe to leave.
- Updating WooCommerce core fixes this The flaw is in the invoice plugin, not WooCommerce itself. The record names the plugin as the vulnerable component; updating WooCommerce alone does not address it.

How we would triage it

This is our judgement.

Step 1: Find the plugin and its version

- In WordPress admin, go to Plugins and search for 'PDF Invoice for WooCommerce'. Note the version shown — the record says all versions up to and including 7.1.1 are affected.
- Confirm whether the plugin is active or only installed; note that the record describes the trigger as an order placement event, which requires the plugin to be active.

Step 2: Disable the plugin

- In WordPress admin, click Deactivate on the plugin. The record names no fixed version, so deactivation is the only available mitigation at this time.
- If you use WP-CLI, deactivate using the slug shown in the previous step's output.

Step 3: Monitor for a patched release

- Check the plugin's WordPress.org listing for a version above 7.1.1. The record names no fixed version; apply only a release the vendor confirms addresses CVE-2026-109220.
- Record the version you were running and the date of deactivation in case a vendor enquiry or future investigation requires it.

Step 4: Confirm deactivation and restore if a patch arrives

- Verify the plugin shows as inactive in your Plugins list and that WooCommerce order processing continues to work without it.
- Do not reactivate on the basis of a version increment alone — wait for a vendor statement that specifically names CVE-2026-109220 as resolved.

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.

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 about a security flaw in a WordPress plugin we may be running: CVE-2026-109220 in the PDF Invoice for WooCommerce + Drag and Drop Template Builder plugin. The record scores it 9.8. All versions up to and including 7.1.1 are affected, and no patched version is recorded. The flaw allows an unauthenticated attacker to run code on the server via the WooCommerce checkout. We need to know whether this plugin is installed, whether it is active, and which version is running. We have disabled the plugin as a precaution. Please confirm in writing the version installed and whether any further action is needed. We would like written confirmation by 12 November 2026.*

The weakness behind it

The record classes this as CWE-502.

The sources, in their words

The CVE record:

> *The PDF Invoice for WooCommerce + Drag and Drop Template Builder plugin for WordPress is vulnerable to PHP Object Injection in all versions up to, and including, 7.1.1 via the 'customer_note' parameter parameter. This is due to insufficient sanitization of the customer_note checkout field, which is inlined verbatim into invoice HTML, entity-decoded back into a live tag by simplehtmldom, reassembled into raw bytes by mPDF, and passed to unserialize() in _getObjAttr without validation. This makes it possible for unauthenticated attackers to execute code on the server. An unauthenticated attacker can trigger the vulnerable code path on a default install by placing an anonymous WooCommerce ...*

Versions of addonsorg PDF Invoice for WooCommerce + Drag and Drop Template Builder the record lists as affected:

- all versions up to and including 7.1.1

The record gives no version ranges. Use the vendor's advisory for the affected versions.

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-2026-109220
```

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-2026-109220
```

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

- No fixed or patched version is named in any source — it is not known when or whether a patch will be released.
- Whether disabling the cheque and direct bank transfer payment methods alone would prevent the attack from reaching the vulnerable code path.
- Whether a web application firewall rule exists that would block the crafted payload before the plugin processes it.
- Exploitation is not recorded. That is not the same as none.
- This is a new record: scores, versions and wording can change in the first days, and they are as at 5 November 2026. We have not read the vendor's advisory page and we have not tested it.

- 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-2026-109220: score 9.8, `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H` As the CVE record gives it.

Metric	Value	What it means here
Attack vector	Network	Reachable over a network, with no need to be on the same segment.
Attack complexity	Low	No special conditions beyond the flaw itself.
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.

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
Plugin active at version ≤ 7.1.1, WooCommerce Store API reachable from the internet	High	Immediately
Plugin active at version ≤ 7.1.1, store reachable by untrusted users on a private network	High	This week
Plugin installed but deactivated, version ≤ 7.1.1	Medium	Next patch cycle
WooCommerce present but this invoice plugin is not installed	None	None for this CVE

Appendix 4. Definition of done

- Every addonsorg PDF Invoice for WooCommerce + Drag and Drop Template Builder you run is listed, with its version.
- Each is outside the affected list, or the vendor's fix or mitigation is in place.
- The version output or a screenshot is saved with the date.
- Anything not done has an owner and a date.

Sources

- CVE record for CVE-2026-109220: https://www.cve.org/CVERecord?id=CVE-2026-109220
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Reference in the CVE record: www.wordfence.com: https://www.wordfence.com/threat-intel/vulnerabilities/id/1bc3529c-6161-4b89-8a35-c3a9b7b8c308?source=cve
- Reference in the CVE record: plugins.trac.wordpress.org: https://plugins.trac.wordpress.org/browser/pdf-for-woocommerce/tags/7.1.1/vendor/mpdf/mpdf/src/Mpdf.php#L9105
- Reference in the CVE record: plugins.trac.wordpress.org: https://plugins.trac.wordpress.org/browser/pdf-for-woocommerce/tags/7.1.1/woocommerce/shortcodes.php#L708