contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

pgadmin.org pgAdmin 4: new CVE

CVE-2026-116241 · record published 30 October 2026

Analysis Report

If you run pgAdmin 4 before 9.19, act now: limit who can sign in to it, and ask your IT provider to move to 9.19 once it is released. No release is confirmed yet. If you do not run pgAdmin 4, stop ...

PREPARED BY	NamiQ	DATE	31 October 2026
FACTS AS OF	31 October 2026	VERSION	1.0
SEVERITY	**Critical: CVSS 3.1 9.9**	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/pgadmin-org-pgadmin-4-cve-2026-116241

The bottom line

> *If you run pgAdmin 4 before 9.19, act now: limit who can sign in to it, and ask your IT provider to move to 9.19 once it is released. No release is confirmed yet. If you do not run pgAdmin 4, stop here.*

At a glance

Fact	Detail
Product	pgadmin.org pgAdmin 4 (CVE-2026-116241)
Severity	Critical: 9.9 (CVSS 3.1)
Status	A new CVE record, published by PostgreSQL on 30 October 2026. Not on CISA's list of exploited vulnerabilities when we read it on 31 October 2026, which means nothing public says yet that anyone is using it.
Fixed in	pgAdmin 4 9.19. The record lists versions before 9.19 as affected. The tracking issue says the fix is pending for 9.19, and no source confirms a release.
Who looks first	Anyone running pgAdmin 4 earlier than 9.19, above all where several people or database roles can reach it. The host running pgAdmin is what is at risk, not only the database server.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run pgAdmin 4 anywhere, including on a laptop or a server?	Nothing to do for this CVE.	Check the version (next question).
Is the version you run earlier than 9.19?	Outside the listed range. Nothing more for this CVE.	You are in the affected range. Go to the next question.
Can anyone you do not fully trust sign in to pgAdmin and use Import/Export on a query?	Lower exposure, but still plan the update to 9.19.	Treat as urgent: reduce access now, then update.

What it is

A user with an ordinary database role can make pgAdmin's Import/Export tool run an operating-system command on the pgAdmin host, as the account running pgAdmin. The record's description is cut off, so it does not state the full impact.

Why it matters

Our reading: the command runs on the pgAdmin host, not on the database server. An ordinary database role is enough, so a low-privilege account can become a foothold on the machine used to manage your databases. The record scores it 9.9 under CVSS 3.1 and names CWE-78 and CWE-115. The fix is not yet released, so exposure reduction is the only step available today.

What the vendor's advisory says

We read one page on github.com. In short:

- The tracking issue was opened on 23 October 2026 as a placeholder. It says details will be published with the pgAdmin 4 9.19 release notes.
- Two commits were added to the issue on 26 October 2026. One rejects a backslash in a double-quoted identifier in Import/Export. The other rejects a backslash or null byte in a schema, table or column name.
- The issue shows no assignee, milestone or linked pull request. It names no workaround and lists no other flaws.

Things people will get wrong

- It is fixed, so we are safe. The tracking issue says the fix is pending for 9.19. Commits exist, but no source confirms a release.
- It needs a database administrator account. The record says an ordinary database role can trigger it.
- It only affects the database server. The record says the command runs on the pgAdmin host, as the account running pgAdmin.
- It is not on CISA's list, so ignore it. The record is new. Not being listed when read says nothing about whether anyone is using it.

How we would triage it

This is our judgement.

The commands only read: they print a version or list what is installed. They come from general knowledge of this kind of product, not from the sources, so check them against the vendor's page before you run them. Words in capitals are for you to fill in.

Step 1: Find where pgAdmin 4 runs

- Ask your IT provider to list every machine and container running pgAdmin 4, including desktop installs.
- Record the version of each. Anything earlier than 9.19 is in the affected range.

```
pip show pgadmin4
docker ps --format '{{.Names}} {{.Image}}'
```

Step 2: Reduce who can reach it

- Limit sign-in to people who need pgAdmin. Remove accounts and database roles that do not.
- Keep it off the public internet where you can. The sources name no workaround, so this is our reading.

Step 3: Update when 9.19 is released

- Watch for the pgAdmin 4 9.19 release notes, which the tracking issue says will carry the details.
- Move every affected install to 9.19 once it is out, then restart pgAdmin.

Step 4: Confirm the result

- Ask for the running version on each install, in writing.
- Confirm that nobody uses the Import/Export tool on queries from untrusted accounts in the meantime.

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.

> *Hello. We would like to know whether we run pgAdmin 4 anywhere, including desktop installs and containers, and which version each one is. CVE-2026-116241 affects versions before 9.19, and the vendor's tracking issue says the fix is pending for 9.19. Please tell us where it runs and who can sign in to it. Please limit access until 9.19 is available, then update. Could you confirm the installed versions and the update in writing by [date]? Thank you.*

The weakness behind it

The record classes this as CWE-78 and CWE-115.

The sources, in their words

The CVE record:

> *pgAdmin 4's Import/Export Data tool wraps a user's query export in a psql \copy (...) TO command and checks that the query's parentheses are balanced so that nothing can escape the wrapper. The check rejected a backslash inside a single-quoted string, but not inside a double-quoted identifier, where it treated the next double quote as closing the identifier. psql's \copy parser applies the connection's escape character to whichever quote opened a token, so with standard_conforming_strings off a backslash-escaped double quote does not close the identifier. pgAdmin's and psql's view of the quoting then diverged, a closing parenthesis hidden from the check ended the \copy (...) wrapper, and a ...*

Versions of pgadmin.org pgAdmin 4 the record lists as affected:

- all versions up to, but not including, 9.19

What we do not know yet

- Whether pgAdmin 4 9.19 has been released. The sources only say the fix is pending for it.
- The full impact. The record's description is cut off after saying an ordinary database role can trigger it.
- Whether any workaround exists. The sources name none.
- Whether anyone is exploiting it. The record is new and CISA did not list it when read.
- 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 31 October 2026. We read the vendor pages listed under Sources; nothing here has been

tested.

- 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-2026-116241: score 9.9, `CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/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	Low	An ordinary account is enough.
User interaction	None	No one has to do anything.
Scope	Changed	Damage reaches beyond 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
30 October 2026	CVE record published by PostgreSQL.

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
pgAdmin before 9.19 that several people or untrusted database roles can reach	Medium	this week
pgAdmin before 9.19 reachable from the internet	Medium	this week; reduce exposure first
pgAdmin before 9.19 used only by one trusted administrator	Low	next patch cycle
pgAdmin is not installed anywhere in your estate	None	none for this CVE

Appendix 4. Definition of done

- Every pgadmin.org pgAdmin 4 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-2026-116241: https://www.cve.org/CVERecord?id=CVE-2026-116241
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Advisory or release notes read: github.com/pgadmin-org/pgadmin4/issues/10663: https://github.com/pgadmin-org/pgadmin4/issues/10663
- Reference in the CVE record: github.com: https://github.com/pgadmin-org/pgadmin4/commit/9be31a019f780647fb689c458408c8803263c812