



# All about Common vulnerabilities and exposures in PostgreSQL

**Priyanka Chatterjee**

# Who am I ?

**A DBA!**

**12 years of experience**

**Past Employers include AWS, Adjust, Groupon,  
Subex**

**Curently member of Postgres Product Team at  
STACKIT**

Let's connect!



STACKIT is the Schwarz Group's cloud and colocation provider



stackit.de



# Agenda



 **The CVE Program and components**

 **CNA**

 **Vulnerability Hunting 101**

 **PostgreSQL & CVE Handling**

 **Bug Reporting & Resolution**

 **Takeaways & Action Steps**



# What is CVE?

Periodic Table of the Elements

1 IA 11A H Hydrogen 1.008	2 IIA 2A He Helium 4.003																
3 Li Lithium 6.941	4 Be Beryllium 9.012											5 B Boron 10.811	6 C Carbon 12.011	7 N Nitrogen 14.007	8 O Oxygen 15.999	9 F Fluorine 18.998	10 Ne Neon 20.180
11 Na Sodium 22.990	12 Mg Magnesium 24.305	3 IIIB 3B	4 IVB 4B	5 VB 5B	6 VIB 6B	7 VIIB 7B	8 VIII 8	9 VIII 8	10 VIII 8	11 IB 1B	12 IIB 2B	13 Al Aluminum 26.982	14 Si Silicon 28.086	15 P Phosphorus 30.974	16 S Sulfur 32.066	17 Cl Chlorine 35.453	18 Ar Argon 39.948
19 K Potassium 39.098	20 Ca Calcium 40.078	21 Sc Scandium 44.956	22 Ti Titanium 47.88	23 V Vanadium 50.942	24 Cr Chromium 51.996	25 Mn Manganese 54.938	26 Fe Iron 55.833	27 Co Cobalt 58.933	28 Ni Nickel 58.693	29 Cu Copper 63.546	30 Zn Zinc 65.39	31 Ga Gallium 69.723	32 Ge Germanium 72.61	33 As Arsenic 74.922	34 Se Selenium 78.09	35 Br Bromine 79.904	36 Kr Krypton 84.80
37 Rb Rubidium 84.468	38 Sr Strontium 87.62	39 Y Yttrium 88.906	40 Zr Zirconium 91.224	41 Nb Niobium 92.906	42 Mo Molybdenum 95.94	43 Tc Technetium 98.907	44 Ru Ruthenium 101.07	45 Rh Rhodium 102.905	46 Pd Palladium 106.42	47 Ag Silver 107.868	48 Cd Cadmium 112.411	49 In Indium 114.818	50 Sn Tin 118.71	51 Sb Antimony 121.760	52 Te Tellurium 127.6	53 I Iodine 126.904	54 Xe Xenon 131.29
55 Cs Cesium 132.905	56 Ba Barium 137.327	57-71 Lanthanide Series	72 Hf Hafnium 178.49	73 Ta Tantalum 180.948	74 W Tungsten 183.85	75 Re Rhenium 186.207	76 Os Osmium 190.23	77 Ir Iridium 192.22	78 Pt Platinum 195.08	79 Au Gold 196.967	80 Hg Mercury 200.59	81 Tl Thallium 204.383	82 Pb Lead 207.2	83 Bi Bismuth 208.980	84 Po Polonium [209]	85 At Astatine 209	86 Rn Radon 222.018
87 Fr Francium 223	88 Ra Radium 226	89-103 Actinide Series	104 Rf Rutherfordium [261]	105 Db Dubnium [262]	106 Sg Seaborgium [266]	107 Bh Bohrium [264]	108 Hs Hassium [269]	109 Mt Meitnerium [268]	110 Ds Darmstadtium [269]	111 Rg Roentgenium [272]	112 Cn Copernicium [277]	113 Uut Ununtrium [288]	114 Fl Flerovium [289]	115 Uup Ununpentium [288]	116 Lv Livermorium [293]	117 Uus Ununseptium [294]	118 Uuo Ununoctium [294]

26  
**Fe**  
Iron  
55.845

57 La Lanthanum 138.905	58 Ce Cerium 140.12	59 Pr Praseodymium 140.908	60 Nd Neodymium 144.24	61 Pm Promethium [145]	62 Sm Samarium 150.36	63 Eu Europium 151.964	64 Gd Gadolinium 157.25	65 Tb Terbium 158.925	66 Dy Dysprosium 162.50	67 Ho Holmium 164.930	68 Er Erbium 167.26	69 Tm Thulium 168.934	70 Yb Ytterbium 173.04	71 Lu Lutetium 174.967
89 Ac Actinium 227	90 Th Thorium 232	91 Pa Protactinium 231	92 U Uranium 238	93 Np Neptunium 237	94 Pu Plutonium 244	95 Am Americium 243	96 Cm Curium 247	97 Bk Berkelium 247	98 Cf Californium 251	99 Es Einsteinium [252]	100 Fm Fermium [257]	101 Md Mendelevium [258]	102 No Nobelium [259]	103 Lr Lawrencium [262]

- Alkali Metal
- Alkaline Earth
- Transition Metal
- Semimetal
- Nonmetal
- Basic Metal
- Halogen
- Noble Gas
- Lanthanide
- Actinide

© 2013 Todd Helmenstein  
chemistry.about.com  
sciencemuseum.org



# CVE list

- Free and publicly available
- Catalog of CVEs :
  - CVE ID : **CVE-yyyy-nnnn**
  - Description :  
    **[Problem Type] [Affected Component] [Cause] [Impact]**
  - at least one public reference



**CVE 25 YEARS** [About](#) [Partner Information](#) [Program Organization](#) [Downloads](#) [Resources & Support](#) [Report/Request](#)

Enter keywords (e.g.: CVE ID, sql injection, etc.)

[Search tips](#) | [Provide feedback](#)

**Notice:** Keyword searching of CVE Records is now available in the search box above. Keywords may include a CVE ID (e.g., CVE-2024-1234), or one or more keywords separated by a space (e.g., authorization, SQL Injection, cross site scripting, etc.). [Learn more here](#).

## CVE® Program Mission

Identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.

Currently, there are **272,332** CVE Records accessible via [Download](#) or [Keyword Search](#) above

The CVE Program partners with community members worldwide to grow CVE content and expand its usage. Click below to learn more about the role of [CVE Numbering Authorities \(CNAs\)](#) and [Roots](#).

[Learn More](#)

[Become a Partner](#)



### News

- [Reminder for CVE Consumers – Please Complete the “CVE Data Usage and Satisfaction Survey” Before April 4, 2025](#)
- [Full Agenda Now Available for CVE/FIRST VulnCon 2025 on April 7-10, 2025!](#)
- [Digi Added as CVE Numbering Authority \(CNA\)](#)
- [Vulnerability Data Enrichment for CVE Records: 250 CNAs on the Enrichment Recognition List for March 25, 2025](#)

[NEWS ICONS](#)

[MORE NEWS](#)

### Events

- [CVE/FIRST VulnCon 2025](#)  
7 April 2025 – 10 April 2025 | Raleigh, North Carolina, USA & Virtual
- [CVE Artificial Intelligence Working Group \(CVEAI WG\) Meeting](#)

<a href="#">Access</a>	<a href="#">Learn</a>	<a href="#">Report/Request</a>
<ul style="list-style-type: none"><li><a href="#">List of Partners</a></li><li><a href="#">CNA Rules</a></li><li><a href="#">CVE Record Lifecycle</a></li><li><a href="#">CVEProject on GitHub for Development</a></li><li><a href="#">Idea tracker</a></li></ul>	<ul style="list-style-type: none"><li><a href="#">About CVE</a></li><li><a href="#">Process</a></li><li><a href="#">Program Organization</a></li><li><a href="#">CVE 25th Anniversary Report</a></li><li><a href="#">Related Efforts</a></li><li><a href="#">Terminology</a></li><li><a href="#">CVE Services for CNAs</a></li></ul>	<ul style="list-style-type: none"><li><a href="#">Report vulnerability/Request CVE ID</a></li><li><a href="#">Request CVE Record be published/updated</a></li><li><a href="#">Report the use of a reserved CVE ID</a></li></ul>

### Access Resources Based on Role



# Looking up for CVE with keywords



Showing 1 - 5 of 5 results for **pgbouncer**

Show: 25 Sort by: CVE ID (new to old)

## **CVE-2025-2291**

CNA: PostgreSQL

Password can be used past expiry in PgBouncer due to `auth_query` not taking into account Postgres its `VALID UNTIL` value, which allows an attacker to log in with an already...

[Show more](#)

## **CVE-2021-3935**

CNA: Fedora Project (Infrastructure Software)

When PgBouncer is configured to use "cert" authentication, a man-in-the-middle attacker can inject arbitrary SQL queries when a connection is first established, despite the use of TLS certificate verification and...

[Show more](#)

## **CVE-2015-6817**

CNA: Debian GNU/Linux

PgBouncer 1.6.x before 1.6.1, when configured with `auth_user`, allows remote attackers to gain login access as `auth_user` via an unknown username.

## **CVE-2015-4054**

CNA: Debian GNU/Linux

PgBouncer before 1.5.5 allows remote attackers to cause a denial of service (NULL pointer dereference and crash) by sending a password packet before a startup packet.

## **CVE-2012-4575**

CNA: Red Hat, Inc.

The `add_database` function in `objects.c` in the pgbouncer pooler 1.5.2 for PostgreSQL allows remote attackers to cause a denial of service (daemon outage) via a long database name in a...



## CVE-2025-2291

### CVE-2025-2291

CNA: PostgreSQL

Password can be used past expiry in PgBouncer due to `auth_query` not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password



## CVE-2025-2291

**CVE-2025-2291**

CNA: PostgreSQL

Password can be used past expiry in PgBouncer due to `auth_query` not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password



## CVE-2025-2291

### CVE-2025-2291

CNA: PostgreSQL

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password



# CVE-2025-2291

Problem type

## CVE-2025-2291

CNA: PostgreSQL

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password



# CVE-2025-2291

Problem Type

Affected Component

CVE-2025-2291

CNA: PostgreSQL

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password



# CVE-2025-2291

Problem Type

Affected Component

Cause

CVE-2025-2291

CNA: PostgreSQL

Password can be used past expiry in PgBouncer due to auth query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password



# CVE-2025-2291

## CVE-2025-2291

Password can be used past expiry in PgBouncer due to auth query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

Problem Type

Affected Component

Cause

CNA: PostgreSQL

Impact

# CVE-2025-2291 details



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

#### CWE 1 Total

[Learn more](#)

- [CWE-324: Use of a Key Past its Expiration Date](#)

#### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

<b>Vendor</b> n/a	<b>Product</b> PgBouncer
----------------------	-----------------------------

#### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before 1.24.1

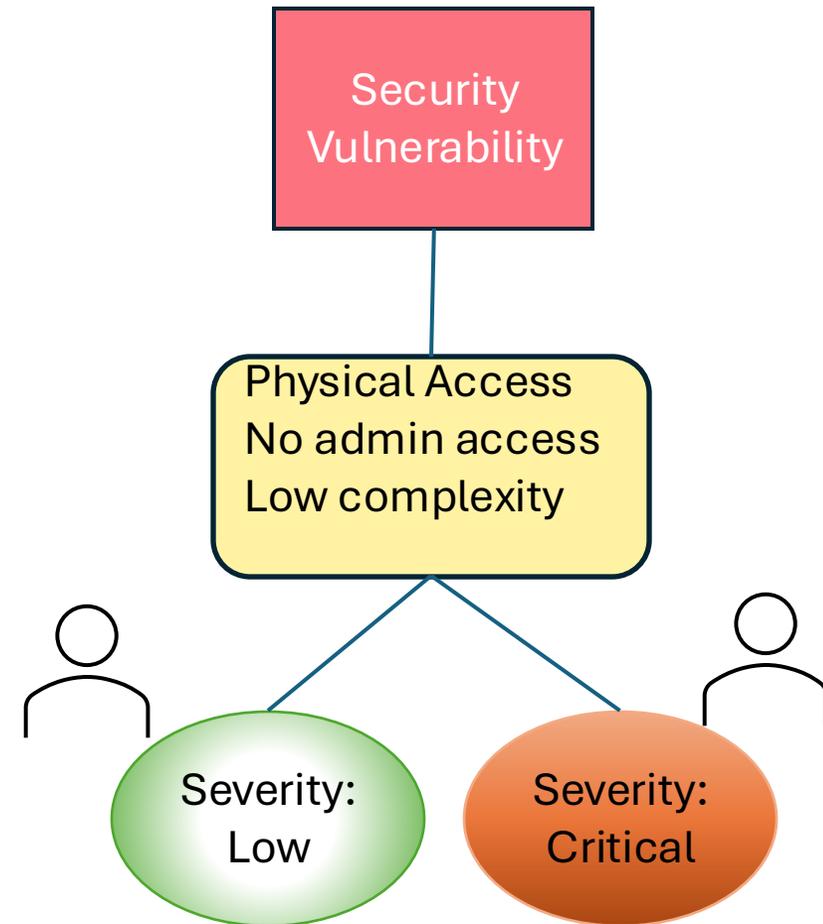
#### References 1 Total

- <https://www.pgбouncer.org/changelog.html#pgбouncer-124x>



# CVSS

- Common Vulnerability Scoring System
- A numerical score to determine the severity of a vulnerability



# CVSS score range



Score Range	Severity
0.0	None
0.1 – 3.9	Low
4.0 – 6.9	Medium
7.0 – 8.9	High
9.0 – 10.0	Critical



# Determining CVSS

Base metric group : mandatory

Temporal metric group

Environmental metric group



# Base metric group

Exploitability metrics	
Attack Vector	: <i>N, A, L, P</i>
Attack Complexity	: <i>L, H</i>
Privileges Required	: <i>N, L, H</i>
User Interaction	: <i>N, R</i>

Impact metrics	
Confidentiality impact	: <i>N, L, H</i>
Integrity impact	: <i>N, L, H</i>
Availability impact	: <i>N, L, H</i>
Scope	: <i>U, C</i>



# Temporal metric group

Explicit code maturity : X, U, P, F, H

Remediation level : X, O, T, W, U

Report confidence : X, U, R, C



# Environmental metric group

## Exploitability Metrics

Attack Vector

Attack Complexity

Privileges Required

User Interaction

Scope

## Impact Metrics

Confidentiality metrics

Integrity Impact

Availability Impact

## Impact Subscore Modifier

Confidentiality Requirement : X, L, M, H

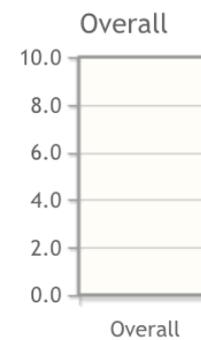
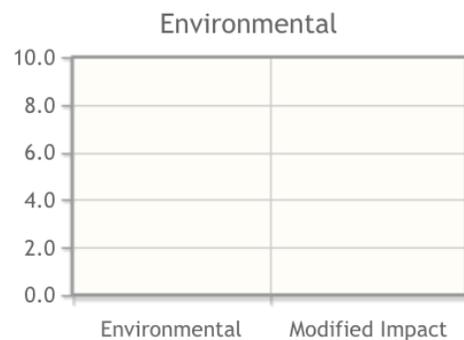
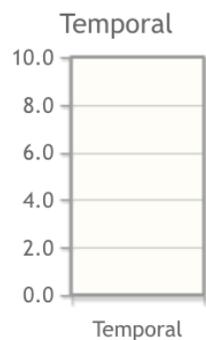
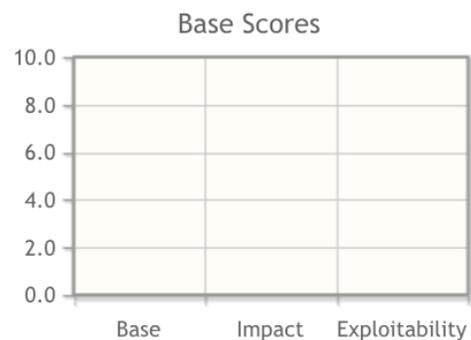
Integrity Requirement : X, L, M, H

Availability Requirement : X, L, M, H



# CVSS calculator

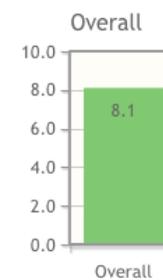
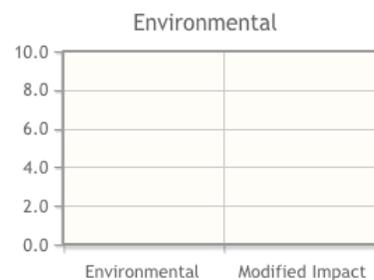
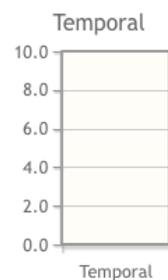
- <https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>



**CVSS Base Score:** NA  
Impact Subscore: NA  
Exploitability Subscore: NA  
**CVSS Temporal Score:** NA  
CVSS Environmental Score: NA  
Modified Impact Subscore: NA  
**Overall CVSS Score:** NA



# CVSS calculator Base Score metrics



**CVSS Base Score: 8.1**  
Impact Subscore: 6.0  
Exploitability Subscore: 1.4  
**CVSS Temporal Score: NA**  
CVSS Environmental Score: NA  
Modified Impact Subscore: NA  
**Overall CVSS Score: 8.1**

Show Equations

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H

## Base Score Metrics

### Exploitability Metrics

#### Attack Vector (AV)\*

Network (AV:N) Adjacent Network (AV:A) **Local (AV:L)** Physical (AV:P)

#### Attack Complexity (AC)\*

Low (AC:L) **High (AC:H)**

#### Privileges Required (PR)\*

**None (PR:N)** Low (PR:L) High (PR:H)

#### User Interaction (UI)\*

**None (UI:N)** Required (UI:R)

### Scope (S)\*

Unchanged (S:U) **Changed (S:C)**

### Impact Metrics

#### Confidentiality Impact (C)\*

None (C:N) Low (C:L) **High (C:H)**

#### Integrity Impact (I)\*

None (I:N) Low (I:L) **High (I:H)**

#### Availability Impact (A)\*

None (A:N) Low (A:L) **High (A:H)**

\* - All base metrics are required to generate a base score.



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



Attack  
vector



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



Attack  
Complexity



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



Privs  
Req



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



User  
interaction



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



Scope



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



Confidentiality



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



Integrity



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H



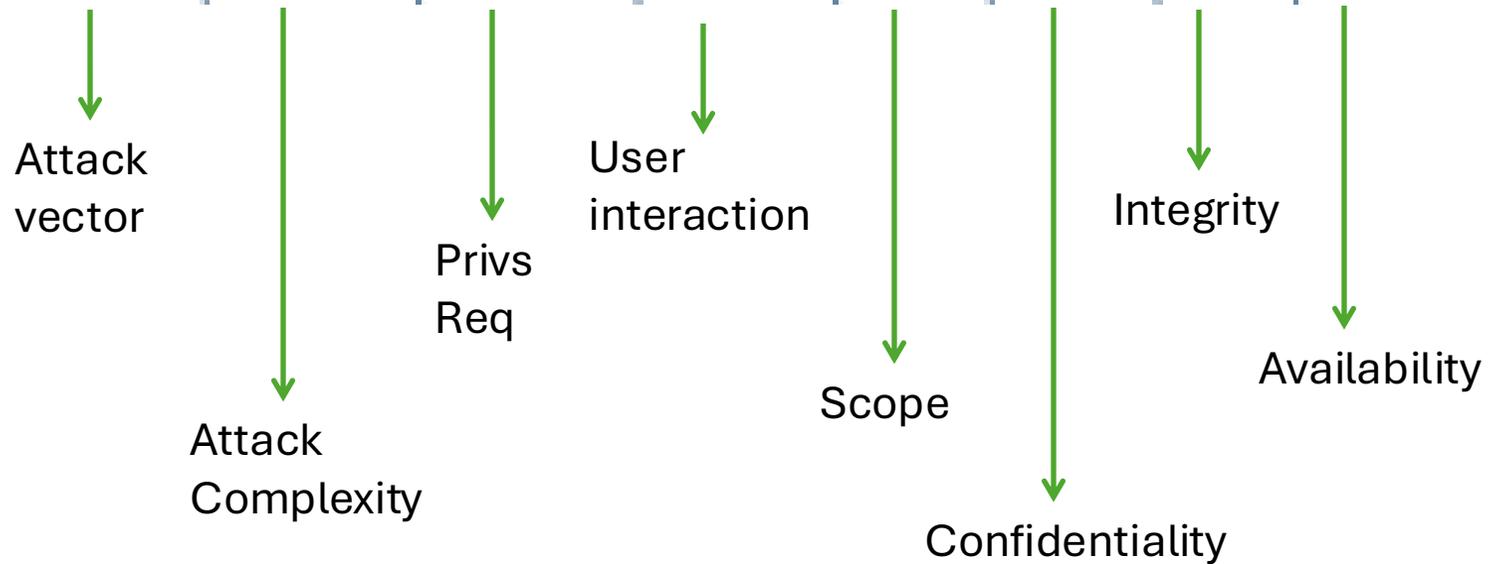
Availability



# CVSS calculator Base Score metrics

## CVSS v3.1 Vector

AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H





# CVSS calculator Temporal and Env. metrics

## Temporal Score Metrics

### Exploit Code Maturity (E)

Not Defined (E:X)  Unproven that exploit exists (E:U)  Proof of concept code (E:P)  Functional exploit exists (E:F)  High (E:H)

### Remediation Level (RL)

Not Defined (RL:X)  Official fix (RL:O)  Temporary fix (RL:T)  Workaround (RL:W)  Unavailable (RL:U)

### Report Confidence (RC)

Not Defined (RC:X)  Unknown (RC:U)  Reasonable (RC:R)  Confirmed (RC:C)

## Environmental Score Metrics

### Exploitability Metrics

#### Attack Vector (MAV)

Not Defined (MAV:X)  Network (MAV:N)  Adjacent Network (MAV:A)  
 Local (MAV:L)  Physical (MAV:P)

#### Attack Complexity (MAC)

Not Defined (MAC:X)  Low (MAC:L)  High (MAC:H)

#### Privileges Required (MPR)

Not Defined (MPR:X)  None (MPR:N)  Low (MPR:L)  High (MPR:H)

#### User Interaction (MUI)

Not Defined (MUI:X)  None (MUI:N)  Required (MUI:R)

#### Scope (MS)

Not Defined (MS:X)  Unchanged (MS:U)  Changed (MS:C)

### Impact Metrics

#### Confidentiality Impact (MC)

Not Defined (MC:X)  None (MC:N)  Low (MC:L)  
 High (MC:H)

#### Integrity Impact (MI)

Not Defined (MI:X)  None (MI:N)  Low (MI:L)  
 High (MI:H)

#### Availability Impact (MA)

Not Defined (MA:X)  None (MA:N)  Low (MA:L)  
 High (MA:H)

### Impact Subscore Modifiers

#### Confidentiality Requirement (CR)

Not Defined (CR:X)  Low (CR:L)  
 Medium (CR:M)  High (CR:H)

#### Integrity Requirement (IR)

Not Defined (IR:X)  Low (IR:L)  Medium (IR:M)  
 High (IR:H)

#### Availability Requirement (AR)

Not Defined (AR:X)  Low (AR:L)  
 Medium (AR:M)  High (AR:H)



# CVSS calculator All metrics

## Temporal Score Metrics

### Exploit Code Maturity (E)

Not Defined (E:X) Unproven that exploit exists (E:U) **Proof of concept code (E:P)** Functional exploit exists (E:F) High (E:H)

### Remediation Level (RL)

Not Defined (RL:X) **Official fix (RL:O)** Temporary fix (RL:T) Workaround (RL:W) Unavailable (RL:U)

### Report Confidence (RC)

Not Defined (RC:X) Unknown (RC:U) **Reasonable (RC:R)** Confirmed (RC:C)

## Environmental Score Metrics

### Exploitability Metrics

#### Attack Vector (MAV)

**Not Defined (MAV:X)** Network (MAV:N) Adjacent Network (MAV:A)  
Local (MAV:L) Physical (MAV:P)

#### Attack Complexity (MAC)

**Not Defined (MAC:X)** Low (MAC:L) High (MAC:H)

#### Privileges Required (MPR)

Not Defined (MPR:X) None (MPR:N) Low (MPR:L) **High (MPR:H)**

#### User Interaction (MUI)

Not Defined (MUI:X) None (MUI:N) **Required (MUI:R)**

#### Scope (MS)

**Not Defined (MS:X)** Unchanged (MS:U) Changed (MS:C)

### Impact Metrics

#### Confidentiality Impact (MC)

**Not Defined (MC:X)** None (MC:N) Low (MC:L)  
High (MC:H)

#### Integrity Impact (MI)

**Not Defined (MI:X)** None (MI:N) Low (MI:L)  
High (MI:H)

#### Availability Impact (MA)

**Not Defined (MA:X)** None (MA:N) Low (MA:L)  
High (MA:H)

### Impact Subscore Modifiers

#### Confidentiality Requirement (CR)

**Not Defined (CR:X)** Low (CR:L)  
Medium (CR:M) High (CR:H)

#### Integrity Requirement (IR)

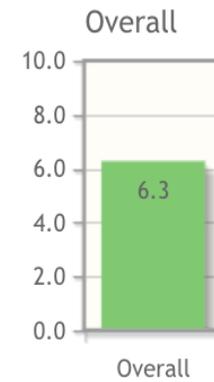
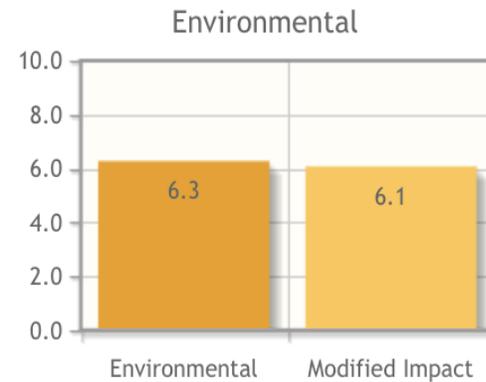
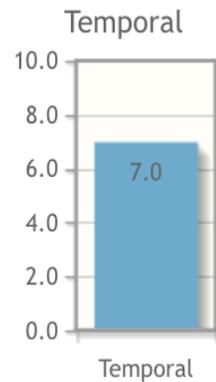
**Not Defined (IR:X)** Low (IR:L) Medium (IR:M)  
High (IR:H)

#### Availability Requirement (AR)

**Not Defined (AR:X)** Low (AR:L)  
Medium (AR:M) High (AR:H)



# CVSS calculator All metrics

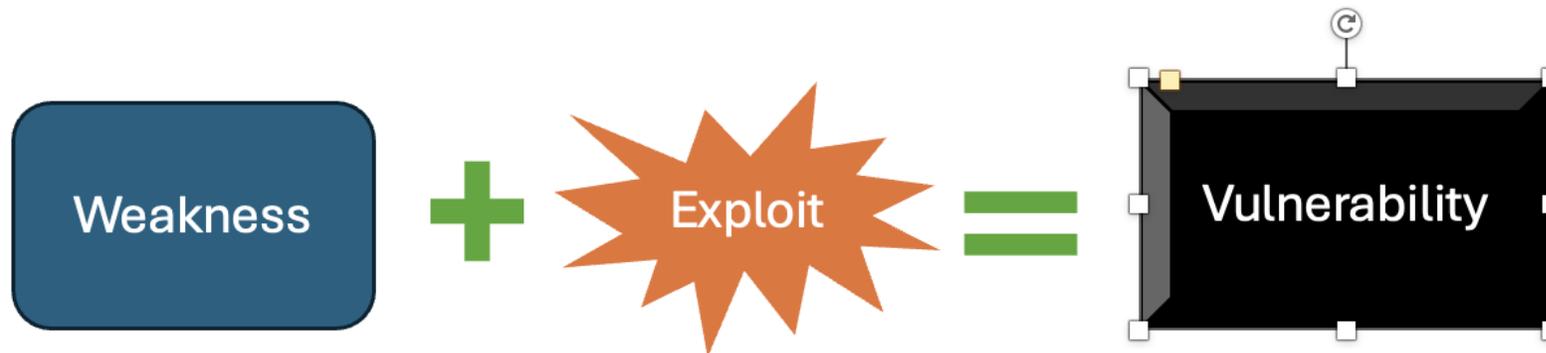
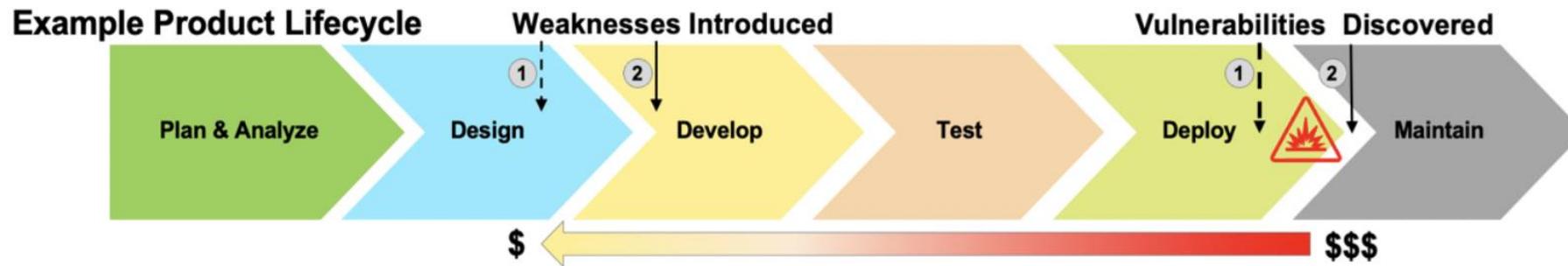


**CVSS Base Score:** 8.1  
Impact Subscore: 6.0  
Exploitability Subscore: 1.4  
**CVSS Temporal Score:** 7.0  
CVSS Environmental Score: 6.3  
Modified Impact Subscore: 6.1  
**Overall CVSS Score:** 6.3



# Common Weakness enumeration

- Community developed list of software and hardware weaknesses.
- Root cause mapping for CVE





# CWE navigation <https://cwe.mitre.org/>

## View CWEs by

Software Development

Hardware Design

All Weaknesses

Other Select Options



# CWE navigation

Software Development

## 699 - Software Development

- API / Function Errors - (1228)
- Audit / Logging Errors - (1210)
- Authentication Errors - (1211)
- Authorization Errors - (1212)
- Bad Coding Practices - (1006)
- Behavioral Problems - (438)
- Business Logic Errors - (840)
- Communication Channel Errors - (417)
- Complexity Issues - (1226)
- Concurrency Issues - (557)
- Credentials Management Errors - (255)
- Cryptographic Issues - (310)
- Key Management Errors - (320)
- Data Integrity Issues - (1214)
- Data Processing Errors - (19)



# CWE navigation

## Software Development

### 699 - Software Development

- [-]  **C** API / Function Errors - (1228)
  - [-]  **B** Use of Inherently Dangerous Function - (242)
  - [-]  **B** Use of Function with Inconsistent Implementations - (474)
  - [-]  **B** Undefined Behavior for Input to API - (475)
  - [-]  **B** Use of Obsolete Function - (477)
  - [-]  **B** Use of Potentially Dangerous Function - (676)
  - [-]  **B** Use of Low-Level Functionality - (695)
  - [-]  **B** Exposed Dangerous Method or Function - (749)
- [-]  **C** Audit / Logging Errors - (1210)
  - [-]  **B** Improper Output Neutralization for Logs - (117)
  - [-]  **B** Truncation of Security-relevant Information - (222)
  - [-]  **B** Omission of Security-relevant Information - (223)
  - [-]  **B** Obscured Security-relevant Information by Alternate Name - (224)
  - [-]  **B** Insufficient Logging - (778)
  - [-]  **B** Logging of Excessive Data - (779)
- [-]  **C** Authentication Errors - (1211)
  - [-]  **B** Authentication Bypass by Alternate Name - (289)
  - [-]  **B** Authentication Bypass by Spoofing - (290)
  - [-]  **B** Authentication Bypass by Capture-replay - (294)
  - [-]  **B** Improper Certificate Validation - (295)
  - [-]  **B** Reflection Attack in an Authentication Protocol - (301)
  - [-]  **B** Incorrect Implementation of Authentication Algorithm - (303)
  - [-]  **B** Authentication Bypass by Primary Weakness - (305)
  - [-]  **B** Missing Authentication for Critical Function - (306)
  - [-]  **B** Improper Restriction of Excessive Authentication Attempts - (307)
  - [-]  **B** Use of Single-factor Authentication - (308)
  - [-]  **B** Use of Password System for Primary Authentication - (309)
  - [-]  **B** Key Exchange without Entity Authentication - (322)
  - [-]  **B** Use of Client-Side Authentication - (603)
  - [-]  **B** Overly Restrictive Account Lockout Mechanism - (645)



# CWE navigation

Software Development

Use of Single-factor Authentication - (308)

## CWE-308: Use of Single-factor Authentication

Weakness ID: 308  
**Vulnerability Mapping: ALLOWED**  
Abstraction: Base

View customized information:

Conceptual

Operational

Mapping  
Friendly

Complete

Custom

### Description

The use of single-factor authentication can lead to unnecessary risk of compromise when compared with the benefits of a dual-factor authentication

### Extended Description

While the use of multiple authentication schemes is simply piling on more complexity on top of authentication, it is inestimably valuable to have such passwords is rampant on the internet. Without the added protection of multiple authentication schemes, a single mistake can result in the compromise and also easy to use, they should be implemented and required.

### Common Consequences

#### Impact Details

*Bypass Protection  
Mechanism*

**Scope: Access Control**

If the secret in a single-factor authentication scheme gets compromised, full authentication is possible.

### Potential Mitigations

Phase(s)	Mitigation
Architecture and Design	Use multiple independent authentication schemes, which ensures that -- if one of the methods is compromised



# Enrichment

- **ADP** (Authorised Data Publisher)
  - Enriches a published CVE
  - + **Review – Research**
  - + **CVSS score**
  - + **CWE**
  - + **Additional information**
- Eg: NVD, CISA, Github Advisory Databases

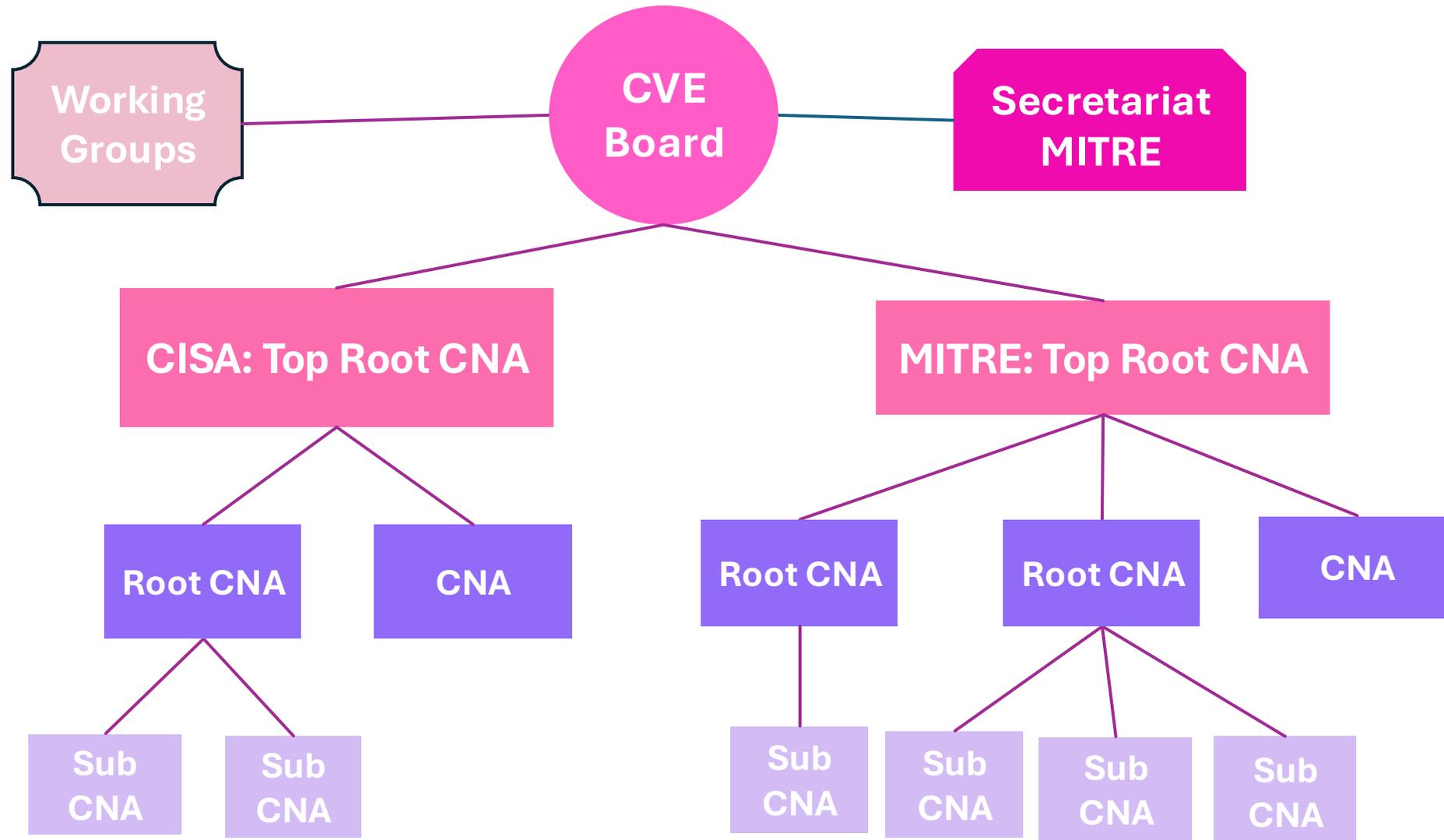


# What is CNA?

- CVE numbering Authority
- Every CNA has a specific scope
- CNAs control CVE publication
- Provide enrichment since April 2024



# CNA organization



# CVE-2025-2291 details recap



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

### CWE 1 Total

[Learn more](#)

- [CWE-324: Use of a Key Past its Expiration Date](#)

### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

<b>Vendor</b> n/a	<b>Product</b> PgBouncer
----------------------	-----------------------------

### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before 1.24.1

### References 1 Total

- <https://www.pgouncer.org/changelog.html#pgbouncer-124x>

# CVE-2025-2291 details recap



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

### CWE 1 Total

[Learn more](#)

- [CWE-324: Use of a Key Past its Expiration Date](#)

### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

Vendor	Product
n/a	PgBouncer

### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before [1.24.1](#)

### References 1 Total

- <https://www.pgouncer.org/changelog.html#pgbouncer-124x>

# CVE-2025-2291 details recap



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

### CWE 1 Total

[Learn more](#)

- [CWE-324: Use of a Key Past its Expiration Date](#)

### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

<b>Vendor</b> n/a	<b>Product</b> PgBouncer
----------------------	-----------------------------

### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before 1.24.1

### References 1 Total

- <https://www.pgouncer.org/changelog.html#pgbouncer-124x>

# CVE-2025-2291 details recap



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

### CWE 1 Total

[Learn more](#)

- **CWE-324: Use of a Key Past its Expiration Date**

### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

<b>Vendor</b> n/a	<b>Product</b> PgBouncer
----------------------	-----------------------------

### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before **1.24.1**

### References 1 Total

- <https://www.pgouncer.org/changelog.html#pgbouncer-124x>

# CVE-2025-2291 details recap



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

### CWE 1 Total

[Learn more](#)

- [CWE-324: Use of a Key Past its Expiration Date](#)

### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

<b>Vendor</b> n/a	<b>Product</b> PgBouncer
----------------------	-----------------------------

### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before 1.24.1

### References 1 Total

- <https://www.pgouncer.org/changelog.html#pgbouncer-124x>

# CVE-2025-2291 details recap



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

### CWE 1 Total

[Learn more](#)

- [CWE-324: Use of a Key Past its Expiration Date](#)

### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

<b>Vendor</b> n/a		<b>Product</b> PgBouncer
----------------------	---	-----------------------------

### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before 1.24.1

### References 1 Total

- <https://www.pgouncer.org/changelog.html#pgbouncer-124x>

# CVE-2025-2291 details recap



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

### CWE 1 Total

[Learn more](#)

- [CWE-324: Use of a Key Past its Expiration Date](#)

### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

<b>Vendor</b> n/a	<b>Product</b> PgBouncer
----------------------	-----------------------------

### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before 1.24.1

### References 1 Total

- <https://www.pgouncer.org/changelog.html#pgbouncer-124x>



# CVE-2025-2291 details recap



## CNA: PostgreSQL

**Published:** 2025-04-16 **Updated:** 2025-04-16  
**Title:** PgBouncer Default Auth\_query Does Not Take Postgres Password Expiry Into Account

### Description

Password can be used past expiry in PgBouncer due to auth\_query not taking into account Postgres its VALID UNTIL value, which allows an attacker to log in with an already expired password

### CWE 1 Total

[Learn more](#)

- [CWE-324: Use of a Key Past its Expiration Date](#)

### CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
8.1	HIGH	3.1	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

### Product Status

[Learn more](#)

<b>Vendor</b> n/a	<b>Product</b> PgBouncer
----------------------	-----------------------------

### Versions 1 Total

*Default Status: unaffected*  
Affected

- affected from 0 before 1.24.1

### References 1 Total

- <https://www.pgouncer.org/changelog.html#pgbouncer-124x>





# Reference example

- Security

- Fix CVE-2025-2291: Previously PgBouncer did not take into account the VALID UNTIL of a user password when querying for password hashes using its auth\_query. So if PgBouncer is used as a transparent proxy in front of Postgres it could allow passwords that had already expired. To solve this issue the default auth\_query and the examples of custom auth\_query functions in the documentation have been changed to take VALID UNTIL into account. If you are using a custom auth\_query you should update that accordingly. If you are using the default auth\_query, you can either update to PgBouncer 1.24.1 or change your config to use the new default auth\_query on a previous release of PgBouncer.

- Fixes

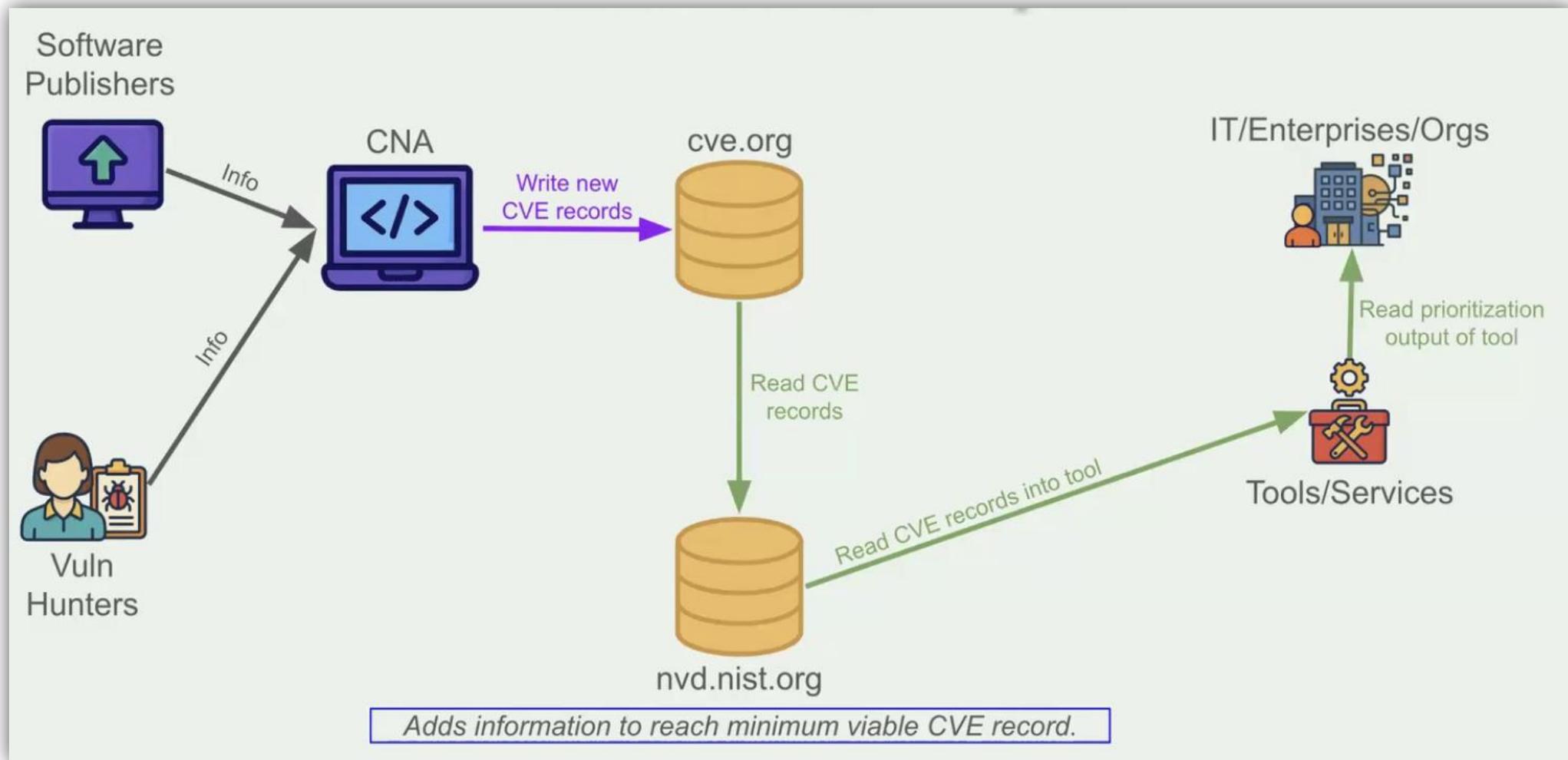
- Fix PAM support by reverting pam authentication support in HBA file. ([#1291](#)) (bug introduced in 1.24.0)
- Fix bug when decrementing user connection count. This was included in the tag of 1.24.0 on GitHub, but the release tarball did not contain this fix. ([#1238](#)) (bug introduced in 1.24.0)
- Add `test_load_balance_hosts.py` to the tarball. ([#1282](#))
- Fix issues with tests to allow them to be run by Debian packagers. ([#1266](#), [#1250](#))

- Docs

- Update `auth_query` example to set a safe `search_path`. ([#1245](#))

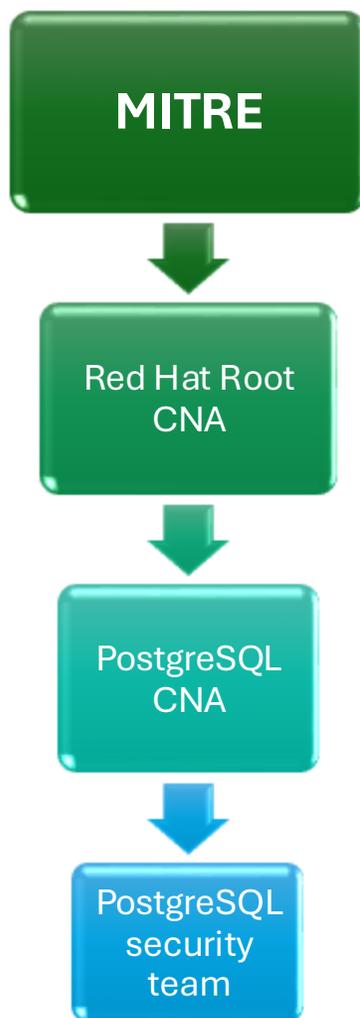


# Current CVE lifecycle





# PostgreSQL and CVEs



## Projects :

- PostgreSQL
- PostgreSQL RPM and DEB packaging
- PostgreSQL windows/mac installers
- pgJDBC
- psqlODBC
- pgAdmin
- pgbouncer



# Reporting a bug

PostgreSQL

PostgreSQL installers

pgAdmin



[security@postgresql.org](mailto:security@postgresql.org)

PostgreSQL JDBC driver



[ppsql-jdbc-security@lists.postgresql.org](mailto:ppgsql-jdbc-security@lists.postgresql.org)

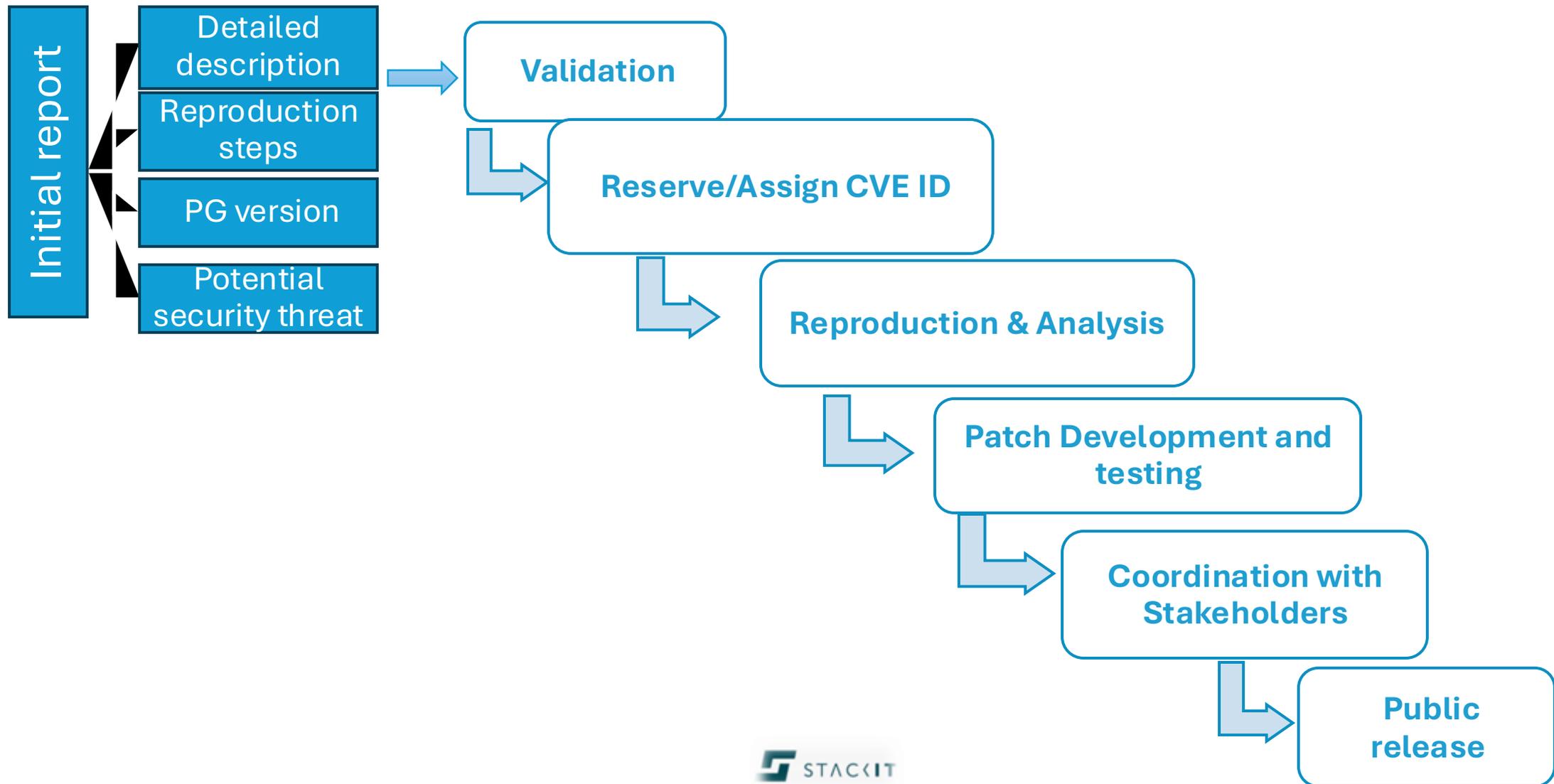
Anything else



[security@postgresql.org](mailto:security@postgresql.org)

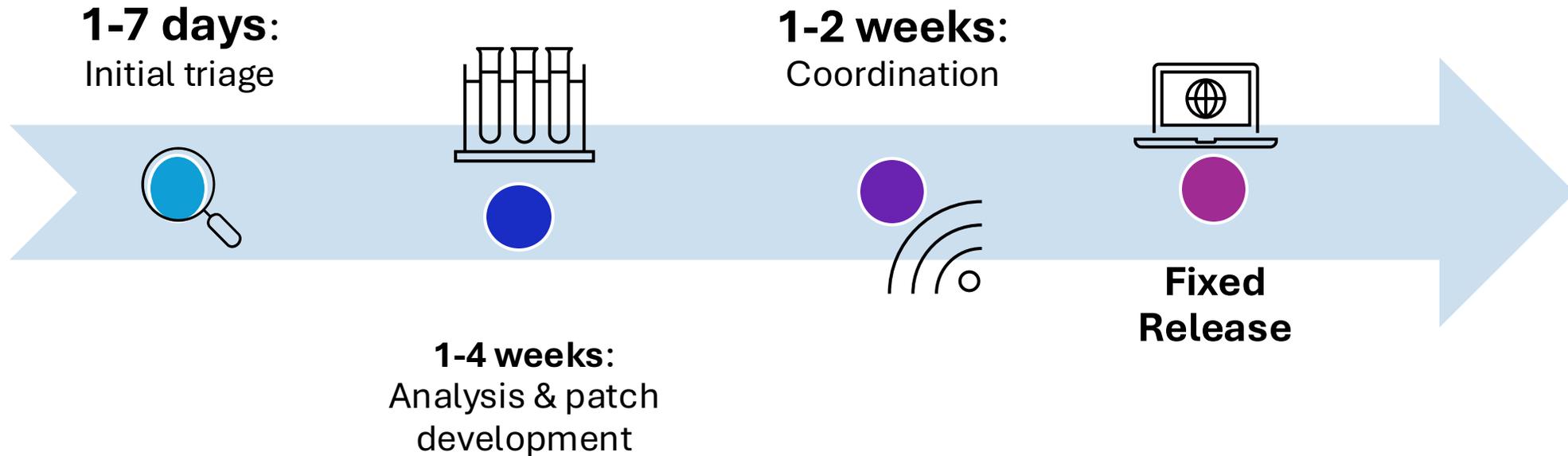


# Lifecycle of a reported bug





# Postgres Security release Timeline(approx)



<https://www.postgresql.org/developer/roadmap/>

At least one minor release every quarter

The second Thursday of February, May, August, and November.

Exception: more releases in between for important bug fixes



# Where to check? What to check?

<https://www.postgresql.org/support/security/>

## Known PostgreSQL Security Vulnerabilities in Supported Versions

You can filter the view of patches to show just patches for version:

[17](#) - [16](#) - [15](#) - [14](#) - [13](#) - [all](#)

Reference	Affected	Fixed	Component & CVSS v3 Base Score	Description
<a href="#">CVE-2025-1094</a> Announcement	17, 16, 15, 14, 13	17.3, 16.7, 15.11, 14.16, 13.19	client <b>8.1</b> AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	PostgreSQL quoting APIs miss neutralizing quoting syntax in text that fails encoding validation <a href="#">more details</a>
<a href="#">CVE-2024-10979</a> Announcement	17, 16, 15, 14, 13	17.1, 16.5, 15.9, 14.14, 13.17	core server <b>8.8</b> AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	PostgreSQL PL/Perl environment variable changes execute arbitrary code <a href="#">more details</a>
<a href="#">CVE-2024-10978</a> Announcement	17, 16, 15, 14, 13	17.1, 16.5, 15.9, 14.14, 13.17	core server <b>4.2</b> AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N	PostgreSQL SET ROLE, SET SESSION AUTHORIZATION reset to wrong user ID <a href="#">more details</a>
<a href="#">CVE-2024-10977</a> Announcement	17, 16, 15, 14, 13	17.1, 16.5, 15.9, 14.14, 13.17	client <b>3.1</b> AV:N/AC:H/PR:N/UI:R/S:U/C:N/I:L/A:N	PostgreSQL libpq retains an error message from man-in-the-middle <a href="#">more details</a>
<a href="#">CVE-2024-10976</a> Announcement	17, 16, 15, 14, 13	17.1, 16.5, 15.9, 14.14, 13.17	core server <b>4.2</b> AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N	PostgreSQL row security below e.g. subqueries disregards user ID changes <a href="#">more details</a>



# Where to check? What to check?

---

<b>CVE-2024-10979</b> Announcement	17, 16, 15, 14, 13 17.1, 16.5, 15.9, 14.14, 13.17	core server <b>8.8</b> AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	PostgreSQL PL/Perl environment variable changes execute arbitrary code <a href="#">more details</a>
---------------------------------------	---	--	--

---



# Where to check? What to check?

**CVE-2024-10979** 17, 16, 15, 14, 13 17.1, 16.5, 15.9, 14.14, 13.17 core server PostgreSQL PL/Perl environment variable changes execute arbitrary code  
**Announcement** **8.8**  
AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H **more details**

↓  
CVE ID

↓  
Major  
Version

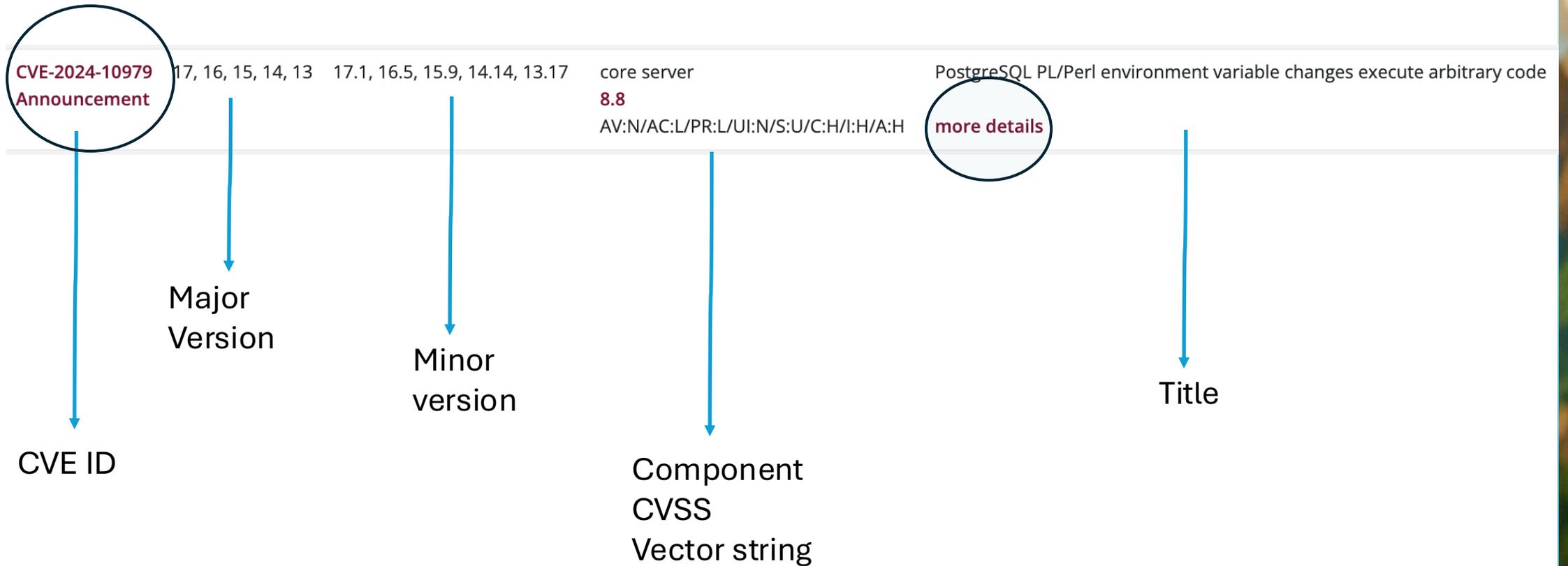
↓  
Minor  
version

↓  
Component  
CVSS  
Vector string

↓  
Title



# Where to check? What to check?





# Where to check? What to check?

CVE-2024-10979  
Announcement

more details

## CVE-2024-10979

### PostgreSQL PL/Perl environment variable changes execute arbitrary code

Incorrect control of environment variables in PostgreSQL *PL/Perl* allows an unprivileged database user to change sensitive process environment variables (e.g. `PATH`). That often suffices to enable arbitrary code execution, even if the attacker lacks a database server operating system user. Versions before PostgreSQL 17.1, 16.5, 15.9, 14.14, 13.17, and 12.21 are affected.

The PostgreSQL project thanks Coby Abrams for reporting this problem.

### Version Information

Affected Version	Fixed In	Fix Published
17	17.1	Nov. 14, 2024
16	16.5	Nov. 14, 2024
15	15.9	Nov. 14, 2024
14	14.14	Nov. 14, 2024
13	13.17	Nov. 14, 2024
12	12.21	Nov. 14, 2024

For more information about [PostgreSQL versioning](#), please visit the [versioning page](#).

### CVSS 3.0

Overall Score	<b>8.8</b>
Component	core server
Vector	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H



# Where to check? What to check?

**CVE-2024-10979** 17, 16, 15, 14, 13 17.1, 16.5, 15.9, 14.14, 13.17 core server  
**Announcement** 8.8  
AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H PostgreSQL PL/Perl environment variable changes execute arbitrary code  
[more details](#)

↓  
CVE ID

↓  
Major  
Version

↓  
Minor  
version

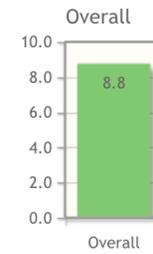
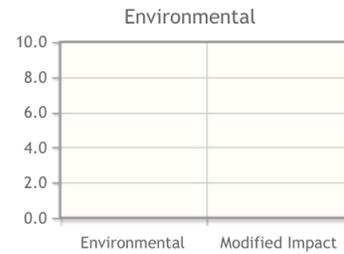
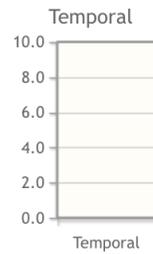
↓  
Component  
CVSS  
Vector string

↓  
Title



# Where to check? What to check?

core server  
**8.8**  
AV:N/AC:L/P



**CVSS Base Score:** 8.8  
 Impact Subscore: 5.9  
 Exploitability Subscore: 2.8  
**CVSS Temporal Score:** NA  
 CVSS Environmental Score: NA  
 Modified Impact Subscore: NA  
**Overall CVSS Score:** 8.8

Show Equations

### CVSS v3.1 Vector

AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

## Base Score Metrics

### Exploitability Metrics

#### Attack Vector (AV)\*

Network (AV:N) Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

#### Attack Complexity (AC)\*

Low (AC:L) High (AC:H)

#### Privileges Required (PR)\*

None (PR:N) Low (PR:L) High (PR:H)

#### User Interaction (UI)\*

None (UI:N) Required (UI:R)

### Scope (S)\*

Unchanged (S:U) Changed (S:C)

### Impact Metrics

#### Confidentiality Impact (C)\*

None (C:N) Low (C:L) High (C:H)

#### Integrity Impact (I)\*

None (I:N) Low (I:L) High (I:H)

#### Availability Impact (A)\*

None (A:N) Low (A:L) High (A:H)



# Vulnerability Hunter....



# Vulnerability Hunting

## Security Issues

**CVE-2025-8713:** PostgreSQL optimizer statistics can expose sampled data within a view, partition, or child table

CVSS v3.1 Base Score: **3.1**

Supported, Vulnerable Versions: 13 - 17.

PostgreSQL optimizer statistics allow a user to read sampled data within a view that the user cannot access. Separately, statistics allow a user to read sampled data PostgreSQL maintains statistics for tables by sampling data available in columns; this data is consulted during the query planning process. Prior to this release, a view access control lists (ACLs) and bypassed row security policies in partitioning or table inheritance hierarchies. Reachable statistics data notably included histogram 2017-7484 and CVE-2019-10130 intended to close this class of vulnerability, but this gap remained. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 The PostgreSQL project thanks Dean Rasheed for reporting this problem.

**CVE-2025-8714:** PostgreSQL `pg_dump` lets superuser of origin server execute arbitrary code in `psql` client

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Untrusted data inclusion in `pg_dump` in PostgreSQL allows a malicious superuser of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands. `pg_dumpall` is also affected. `pg_restore` is affected when used to generate a plain-format dump. This is similar to MySQL CVE-2024-21096. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Martin Rakhmanov, Matthieu Denais, and RyotaK for reporting this problem.

**CVE-2025-8715:** PostgreSQL `pg_dump` newline in object name executes arbitrary code in `psql` client and in restore target server

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Improper neutralization of newlines in `pg_dump` in PostgreSQL allows a user of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands inside a purpose-crafted object name. The same attacks can achieve SQL injection as a superuser of the restore target server. `pg_dumpall`, `pg_restore`, and `pg_upgrade` are also affected. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected. Versions before 11.20 are unaffected. CVE-2012-0868 had fixed this class of problem, but version 11.20 reintroduced it.

The PostgreSQL project thanks Noah Misch for reporting this problem.



# Vulnerability Hunting



## Security Issues

### **CVE-2025-8713:** PostgreSQL optimizer statistics can expose sampled data within a view, partition, or child table

CVSS v3.1 Base Score: **3.1**

Supported, Vulnerable Versions: 13 - 17.

PostgreSQL optimizer statistics allow a user to read sampled data within a view that the user cannot access. Separately, statistics allow a user to read sampled data that a row security policy intended to hide. PostgreSQL maintains statistics for tables by sampling data available in columns; this data is consulted during the query planning process. Prior to this release, a user could craft a leaky operator that bypassed view access control lists (ACLs) and bypassed row security policies in partitioning or table inheritance hierarchies. Reachable statistics data notably included histograms and most-common-values lists. CVE-2017-7484 and CVE-2019-10130 intended to close this class of vulnerability, but this gap remained. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Dean Rasheed for reporting this problem.

### **CVE-2025-8714:** PostgreSQL `pg_dump` lets superuser of origin server execute arbitrary code in `psql` client

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Untrusted data inclusion in `pg_dump` in PostgreSQL allows a malicious superuser of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands. `pg_dumpall` is also affected. `pg_restore` is affected when used to generate a plain-format dump. This is similar to MySQL CVE-2024-21096. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Martin Rakhmanov, Matthieu Denais, and RyotaK for reporting this problem.

### **CVE-2025-8715:** PostgreSQL `pg_dump` newline in object name executes arbitrary code in `psql` client and in restore target server

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Improper neutralization of newlines in `pg_dump` in PostgreSQL allows a user of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands inside a purpose-crafted object name. The same attacks can achieve SQL injection as a superuser of the restore target server. `pg_dumpall`, `pg_restore`, and `pg_upgrade` are also affected. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected. Versions before 11.20 are unaffected. CVE-2012-0868 had fixed this class of problem, but version 11.20 reintroduced it.

The PostgreSQL project thanks Noah Misch for reporting this problem.

# Vulnerability Hunting



## Security Issues

### **CVE-2025-8713:** PostgreSQL optimizer statistics can expose sampled data within a view, partition, or child table

CVSS v3.1 Base Score: **3.1**

Supported, Vulnerable Versions: 13 - 17.

PostgreSQL optimizer statistics allow a user to read sampled data within a view that the user cannot access. Separately, statistics allow a user to read sampled data that a row security policy intended to hide. PostgreSQL maintains statistics for tables by sampling data available in columns; this data is consulted during the query planning process. Prior to this release, a user could craft a leaky operator that bypassed view access control lists (ACLs) and bypassed row security policies in partitioning or table inheritance hierarchies. Reachable statistics data notably included histograms and most-common-values lists. CVE-2017-7484 and CVE-2019-10130 intended to close this class of vulnerability, but this gap remained. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Dean Rasheed for reporting this problem.

### **CVE-2025-8714:** PostgreSQL `pg_dump` lets superuser of origin server execute arbitrary code in `psql` client

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Untrusted data inclusion in `pg_dump` in PostgreSQL allows a malicious superuser of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands. `pg_dumpall` is also affected. `pg_restore` is affected when used to generate a plain-format dump. This is similar to MySQL CVE-2024-21096. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Martin Rakhmanov, Matthieu Denais, and RyotaK for reporting this problem.

### **CVE-2025-8715:** PostgreSQL `pg_dump` newline in object name executes arbitrary code in `psql` client and in restore target server

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Improper neutralization of newlines in `pg_dump` in PostgreSQL allows a user of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands inside a purpose-crafted object name. The same attacks can achieve SQL injection as a superuser of the restore target server. `pg_dumpall`, `pg_restore`, and `pg_upgrade` are also affected. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected. Versions before 11.20 are unaffected. CVE-2012-0868 had fixed this class of problem, but version 11.20 reintroduced it.

The PostgreSQL project thanks Noah Misch for reporting this problem.

# Vulnerability Hunting



## Security Issues

### **CVE-2025-8713:** PostgreSQL optimizer statistics can expose sampled data within a view, partition, or child table

CVSS v3.1 Base Score: **3.1**

Supported, Vulnerable Versions: 13 - 17.

PostgreSQL optimizer statistics allow a user to read sampled data within a view that the user cannot access. Separately, statistics allow a user to read sampled data that a row security policy intended to hide. PostgreSQL maintains statistics for tables by sampling data available in columns; this data is consulted during the query planning process. Prior to this release, a user could craft a leaky operator that bypassed view access control lists (ACLs) and bypassed row security policies in partitioning or table inheritance hierarchies. Reachable statistics data notably included histograms and most-common-values lists. CVE-2017-7484 and CVE-2019-10130 intended to close this class of vulnerability, but this gap remained. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Dean Rasheed for reporting this problem.

### **CVE-2025-8714:** PostgreSQL `pg_dump` lets superuser of origin server execute arbitrary code in `psql` client

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Untrusted data inclusion in `pg_dump` in PostgreSQL allows a malicious superuser of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands. `pg_dumpall` is also affected. `pg_restore` is affected when used to generate a plain-format dump. This is similar to MySQL CVE-2024-21096. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Martin Rakhmanov, Matthieu Denais, and RyotaK for reporting this problem.

### **CVE-2025-8715:** PostgreSQL `pg_dump` newline in object name executes arbitrary code in `psql` client and in restore target server

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Improper neutralization of newlines in `pg_dump` in PostgreSQL allows a user of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands inside a purpose-crafted object name. The same attacks can achieve SQL injection as a superuser of the restore target server. `pg_dumpall`, `pg_restore`, and `pg_upgrade` are also affected. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected. Versions before 11.20 are unaffected. CVE-2012-0868 had fixed this class of problem, but version 11.20 reintroduced it.

The PostgreSQL project thanks Noah Misch for reporting this problem.

# CVSS score range



Score Range	Severity
0.0	None
0.1 – 3.9	Low
4.0 – 6.9	Medium
7.0 – 8.9	High
9.0 – 10.0	Critical

# Vulnerability Hunting

## Security Issues

**CVE-2025-8713:** PostgreSQL optimizer statistics can expose sampled data within a view, partition, or child table

CVSS v3.1 Base Score: **3.1**

Supported, Vulnerable Versions: 13 - 17.

PostgreSQL optimizer statistics allow a user to read sampled data within a view that the user cannot access. Separately, statistics allow a user to read sampled data PostgreSQL maintains statistics for tables by sampling data available in columns; this data is consulted during the query planning process. Prior to this release, a view access control lists (ACLs) and bypassed row security policies in partitioning or table inheritance hierarchies. Reachable statistics data notably included histogram 2017-7484 and CVE-2019-10130 intended to close this class of vulnerability, but this gap remained. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 The PostgreSQL project thanks Dean Rasheed for reporting this problem.

**CVE-2025-8714:** PostgreSQL `pg_dump` lets superuser of origin server execute arbitrary code in `psql` client

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Untrusted data inclusion in `pg_dump` in PostgreSQL allows a malicious superuser of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands. `pg_dumpall` is also affected. `pg_restore` is affected when used to generate a plain-format dump. This is similar to MySQL CVE-2024-21096. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Martin Rakhmanov, Matthieu Denais, and RyotaK for reporting this problem.

**CVE-2025-8715:** PostgreSQL `pg_dump` newline in object name executes arbitrary code in `psql` client and in restore target server

CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Improper neutralization of newlines in `pg_dump` in PostgreSQL allows a user of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands inside a purpose-crafted object name. The same attacks can achieve SQL injection as a superuser of the restore target server. `pg_dumpall`, `pg_restore`, and `pg_upgrade` are also affected. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected. Versions before 11.20 are unaffected. CVE-2012-0868 had fixed this class of problem, but version 11.20 reintroduced it.

The PostgreSQL project thanks Noah Misch for reporting this problem.



# Vulnerability Hunting



## Security Issues

### **CVE-2025-8713:** PostgreSQL optimizer statistics can expose sampled data within a view, partition, or child table

CVSS v3.1 Base Score: **3.1**

Supported, Vulnerable Versions: 13 - 17.

PostgreSQL optimizer statistics allow a user to read sampled data within a view that the user cannot access. Separately, statistics allow a user to read sampled data that a row security policy intended to hide. PostgreSQL maintains statistics for tables by sampling data available in columns; this data is consulted during the query planning process. Prior to this release, a user could craft a leaky operator that bypassed view access control lists (ACLs) and bypassed row security policies in partitioning or table inheritance hierarchies. Reachable statistics data notably included histograms and most-common-values lists. CVE-2017-7484 and CVE-2019-10130 intended to close this class of vulnerability, but this gap remained. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Dean Rasheed for reporting this problem.

### **CVE-2025-8714:** PostgreSQL `pg_dump` lets superuser of origin server execute arbitrary code in `psql` client

CVSS v3.1 Base Score: **8.8**

Supported, vulnerable Versions: 13 - 17.

Untrusted data inclusion in `pg_dump` in PostgreSQL allows a malicious superuser of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands. `pg_dumpall` is also affected. `pg_restore` is affected when used to generate a plain-format dump. This is similar to MySQL CVE-2024-21096. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Martin Rakhmanov, Matthieu Denais, and RyotaK for reporting this problem.

### **CVE-2025-8715:** PostgreSQL `pg_dump` newline in object name executes arbitrary code in `psql` client and in restore target server

CVSS v3.1 Base Score: **8.8**

Supported, vulnerable Versions: 13 - 17.

Improper neutralization of newlines in `pg_dump` in PostgreSQL allows a user of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands inside a purpose-crafted object name. The same attacks can achieve SQL injection as a superuser of the restore target server. `pg_dumpall`, `pg_restore`, and `pg_upgrade` are also affected. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected. Versions before 11.20 are unaffected. CVE-2012-0868 had fixed this class of problem, but version 11.20 reintroduced it.

The PostgreSQL project thanks Noah Misch for reporting this problem.

# Vulnerability Hunting



**CVE-2025-8714:** PostgreSQL `pg_dump` lets superuser of origin server execute arbitrary code in `psql` client

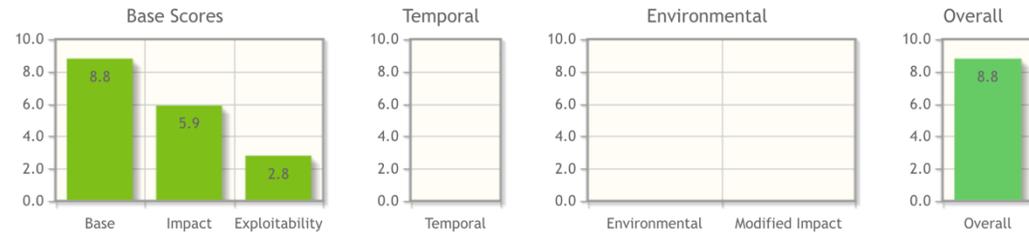
CVSS v3.1 Base Score: **8.8**

Supported, Vulnerable Versions: 13 - 17.

Untrusted data inclusion in `pg_dump` in PostgreSQL allows a malicious superuser of the origin server to inject arbitrary code for restore-time execution as the client operating system account running `psql` to restore the dump, via `psql` meta-commands. `pg_dumpall` is also affected. `pg_restore` is affected when used to generate a plain-format dump. This is similar to MySQL CVE-2024-21096. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.

The PostgreSQL project thanks Martin Rakhmanov, Matthieu Denais, and RyotaK for reporting this problem.

# Vulnerability Hunting



**CVSS Base Score: 8.8**  
 Impact Subscore: 5.9  
 Exploitability Subscore: 2.8  
**CVSS Temporal Score: NA**  
 CVSS Environmental Score: NA  
 Modified Impact Subscore: NA  
**Overall CVSS Score: 8.8**

Show Equations

**CVSS v3.1 Vector**  
 AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

## Base Score Metrics

### Exploitability Metrics

#### Attack Vector (AV)\*

Network (AV:N)
  Adjacent Network (AV:A)
  Local (AV:L)
  Physical (AV:P)

#### Attack Complexity (AC)\*

Low (AC:L)
  High (AC:H)

#### Privileges Required (PR)\*

None (PR:N)
  Low (PR:L)
  High (PR:H)

#### User Interaction (UI)\*

None (UI:N)
  Required (UI:R)

### Scope (S)\*

Unchanged (S:U)
  Changed (S:C)

### Impact Metrics

#### Confidentiality Impact (C)\*

None (C:N)
  Low (C:L)
  High (C:H)

#### Integrity Impact (I)\*

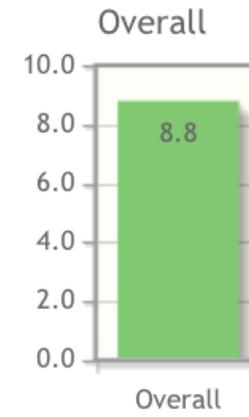
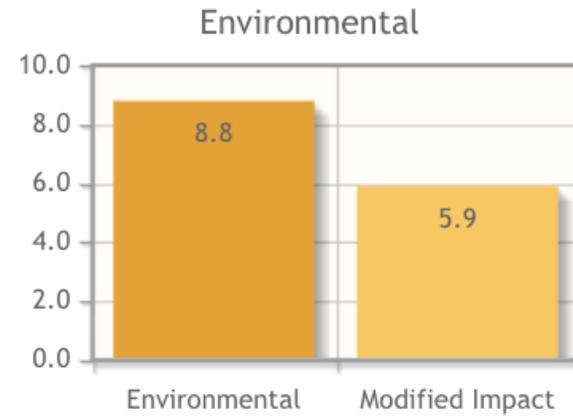
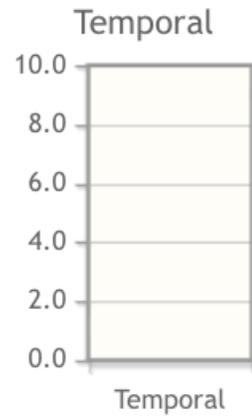
None (I:N)
  Low (I:L)
  High (I:H)

#### Availability Impact (A)\*

None (A:N)
  Low (A:L)
  High (A:H)

\* - All base metrics are required to generate a base score.

# Vulnerability Hunting





# Lets Patch....





# Time to patch

New release

Window of  
Vulnerability





# Integration and automation

- Collaboration with cyber security team/network team/platform team
- Pro actively create internal and external policies
- Runbooks
- Scanning Tools



# Guidelines for a DBA for proactive security measures

**Zero trust principle** : “Never trust, always verify”

- pg\_hba.conf
- least privileged users
- encryptions
- pg\_audit extension
- automated testing
- Improve monitoring and alerting regularly

# Why Should you care?



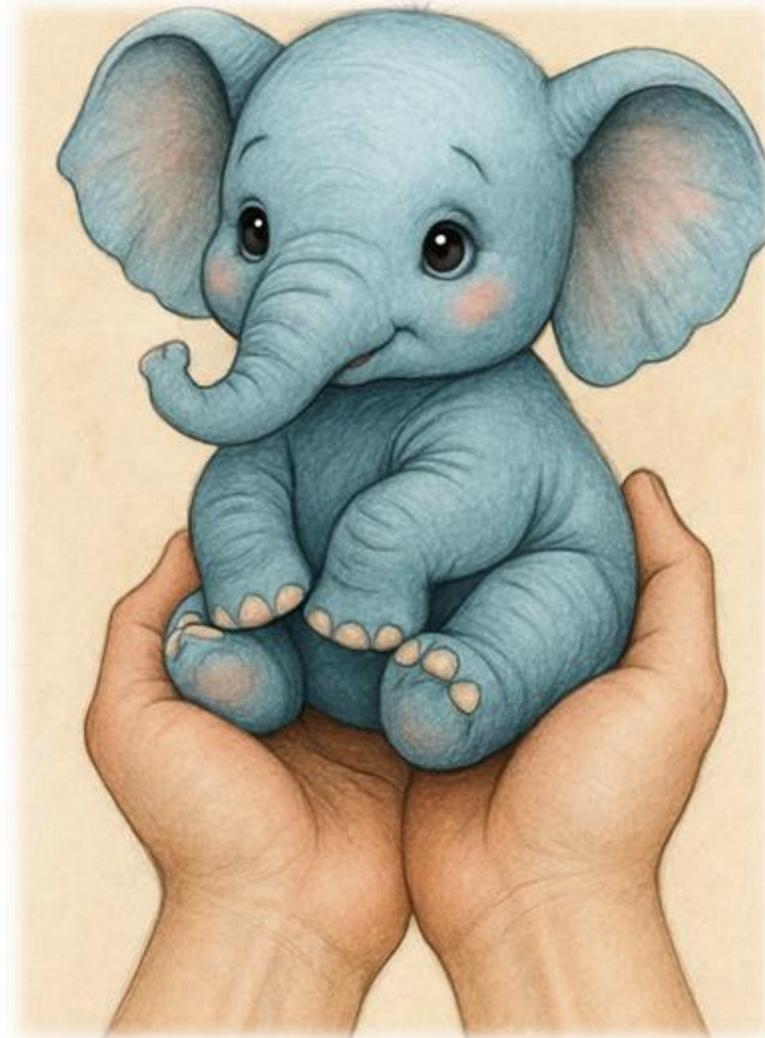
Compliance

Protection

Uptime

Stability

Reputation



**Enable Notification**

<https://www.postgresql.org/list/>  
<https://lists.postgresql.org/>

**Monitor Review Patch**

Feedback



Thank You !