

golang / vulndb Public[Code](#) [Issues 59](#) [Pull requests 5](#) [Actions](#) [Security and quality](#) [Insights](#)[New issue](#)

x/vulndb: potential Go vuln in github.com/jackc/pgproto3 #4518

Closed

Labels

Direct External Report

athuljayaram opened on Feb 19



Acknowledgement

- The maintainer(s) of the affected project have already been made aware of this vulnerability.

Description

`DataRow.Decode` in `github.com/jackc/pgproto3/v2` panics with `slice bounds out of range` when a server sends a `DataRow` message containing a field length that is a negative int32 value (any uint32 in the range `0x80000000 - 0xFFFFFFFF`).

The field length is read correctly as a signed int32:

```
msgSize := int(int32(binary.BigEndian.Uint32(src[rp:])))
```



However, the subsequent bounds check:

```
if len(src[rp:]) < msgSize { ... }
```



is vacuously false when `msgSize` is negative, because `len()` always returns a non-negative integer. The check is bypassed entirely, and the slice expression:

```
dst.Values[i] = src[rp : rp+msgSize : rp+msgSize]
```



panics at runtime. The null sentinel `-1` is handled correctly, but the remaining ~2 billion negative values (`-2` through `-2,147,483,648`) are not guarded.

A malicious or compromised PostgreSQL server can crash any Go application using this library by sending a single crafted `DataRow` message.

Affected Modules, Packages, Versions and Symbols

```
Module: github.com/jackc/pgproto3/v2
Package: github.com/jackc/pgproto3/v2
Versions:
  - Introduced: 2.0.0
  - Fixed: not yet fixed
Symbols:
  - DataRow.Decode
```



CVE/GHSA ID

No response

Fix Commit or Pull Request

No response

References

- [🔒 Negative field length panics in DataRow.Decode jackc/pgx#2507](#)
- <https://securityinfinity.com/research/memory-safety-vulnerabilities-in-go-postgresql-wire-protocol-parsers-pgproto3-pgx>

Additional information

The same pattern was previously identified and correctly patched in a downstream consumer of this library - the fix there uses the guard

```
|| valueLen < 0
```

 alongside the existing bounds check. That fix was never backported to the upstream `pgproto3/v2` library.

CVSS 3.1: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H — 7.5 High

CWE-129: Improper Validation of Array Index

 **athuljayaram** added **Direct External Re...** on Feb 19

 **gopherbot** 3 weeks ago Contributor ...

Change <https://go.dev/cl/755163> mentions this issue: data/reports: add G0-2026-4518

  **gopherbot** closed this as completed in [0b54281](#) 3 weeks ago

  **Ali-Razmjoo** added a commit that references this issue 3 weeks ago

data/reports: add G0-2026-4518 ...

 0b54281

  **GoVulnBot** mentioned this in 2 issues 2 weeks ago

 [x/vulndb: potential Go vuln in github.com/jackc/pgproto3/v2: GHSA-jqcq-xjh3-6g23 #4741](#)

 [x/vulndb: potential Go vuln in github.com/jackc/pgproto3/v2: GHSA-x6gf-mpr2-68h6 #4787](#)

 **gopherbot** 2 weeks ago Contributor ...

Change <https://go.dev/cl/758122> mentions this issue: data/reports: update 4 reports




  **gopherbot** added a commit that references this issue last week

data/reports: update 4 reports ...

861eb2e

 **gopherbot** 3 days ago Contributor ...

Change <https://go.dev/cl/762340> mentions this issue: data/reports: review G0-2026-4518

  **jitsu-net** marked  [x/vulndb: potential Go vuln in github.com/jackc/pgproto3/v2: GHSA-jqcq-xjh3-6g23 #4741](#) as a duplicate of this issue 3 days ago


  **gopherbot** added a commit that references this issue 3 days ago

data/reports: review G0-2026-4518 ...

6cc2989



 **coderabbitai** mentioned this [2 days ago](#)

 [NO-JIRA: no-op change to verify e2e-agent-compact-ipv4 test status on... openshift/assisted-installer#2067](#)

[Sign up for free](#) to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

Metadata

Assignees

No one assigned

Labels

Direct External Report

Type

No type

Projects

No projects


Milestone

No milestone

Relationships

None yet

Development

 Code with agent mode

No branches or pull requests

Participants



