New issue

Negative field length panics in DataRow.Decode #2507

Closed

athuljayaram opened on Feb 19 · edited by athuljayaram Edits

Package: `github.com/jackc/pgproto3` v2.3.3
File: `data_row.go:48-59`
Severity: 7.5 High
CWE: CWE-129 - Improper Validation of Array Index

Summary

`DataRow.Decode` panics with `slice bounds out of range` when a server sends a field length that is a negative int32 value (any uint32 in the range `0x80000000` – `0xFFFFFFFE`).

Root Cause

Line 48 reads the field length correctly as a signed int32:

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

The null sentinel check on the following line correctly handles -1.
However, the subsequent bounds check at line 55:

if len(src[rp:]) < msgSize {
return &invalidMessageFormatErr{messageType: "DataRow"}
}

is vacuously false whenever msgSize is negative. In Go, len() always returns a non-negative integer, so len(...) < -13619152 can never be true. The guard is bypassed entirely, and execution reaches line 59:

```
dst.Values[i] = src[rp : rp+msgSize : rp+msgSize]
// panic: runtime error: slice bounds out of range [:-13619152]
```

The process terminates unconditionally.

Impact

A malicious or compromised PostgreSQL server can crash any Go application using this library by sending a single crafted DataRow message. No special privilege is required — any server the client connects to can trigger this.

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

Fix

Add a msgSize < 0 guard between the null check and the bounds check:

```
if msgSize == -1 {
dst.Values[i] = nil
} else if msgSize < 0 || len(src[rp:]) < msgSize {
return &invalidMessageFormatErr{messageType: "DataRow"}
} else {
dst.Values[i] = src[rp : rp+msgSize : rp+msgSize]
rp += msgSize
}
```

Reference

https://securityinfinity.com/research/memory-safety-vulnerabilities-in-go-postgresql-wire-protocol-parsers-pgproto3-pgx

athuljayaram changed the title ~~Negative field length panics in DataRow.Decode (slice bounds out of range)~~ Negative field length panics in DataRow.Decode on Feb 19

athuljayaram mentioned this on Feb 19

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

jackc on Feb 20 Owner

The package you are referring, https://github.com/jackc/pgproto3, to reached end-of-life on July 1, 2025. The current version github.com/jackc/pgx/v5/pgproto3 is not vulnerable to this issue. See

pgx/pgproto3/data_row.go
Lines 48 to 59 in `0be0344`

```
48            // null
49            if valueLen == -1 {
50                    dst.Values[i] = nil
51            } else {
52                    if len(src[rp:]) < valueLen || valueLen < 0 {
53                            return &invalidMessageFormatErr{messageType: "DataRow"}
54                    }
55
56                    dst.Values[i] = src[rp : rp+valueLen : rp+valueLen]
57                    rp += valueLen
58            }
59    }
```

.

athuljayaram on Feb 21 · edited by athuljayaram Edits Author

Reported this issue only due to this

https://github.com/jackc/pgproto3/blob/bc041643406d711f989a8f7ebe7b5a6fff2e29fe/README.md?plain=1#L7C49-L7C102

jackc on Feb 21 Owner

Ah. Sorry. I can see now how that could be misleading. That was written 6 months before the EOL date. It was intended to be read as "only security updates will be accepted from now on up until the project is EOL".

👍 3

GoVulnBot mentioned this 2 weeks ago

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

nancynh mentioned this 2 weeks ago

fix: drop support for EOL pgxv4 driver GoogleCloudPlatform/alloydb-go-connector#767

mattdowdell mentioned this 2 weeks ago

Vulnerability in archived dependency github.com/jackc/pgproto3/v2 go-jet/jet#566

GoVulnBot mentioned this 2 weeks ago

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

jackc closed this as completed 2 weeks ago

coderabbitai mentioned this 2 weeks ago

chore: bump duty to v1.0.1242 flanksource/kopper#21

jackc mentioned this 2 weeks ago

Discovered DoS vulnerability in pgproto3/v2: panic on negative field length (GHSA-jqcq-xjh3-6g23) #2522

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants