New issue

Jump to bottom

some vulnerability - 0x03 an out-of-bound vulnerability in readTextWithDescrFrame function #79

Closed **Jayl1n** opened this issue on Nov 19, 2020 · 7 comments

Jayl1n commented on Nov 19, 2020 • edited

This is the third vulnerability in id3v2frames.go

In readTextWithDescrFrame function, you don't check the size of b , program will happen panic when the size of b is 2 or less than 2 .

testcase 8eff69ad26a59a05ec11e38f3ca6c592f08dcc54.zip

```
panic: runtime error: slice bounds out of range [:3] with capacity 2

goroutine 1 [running]:
github.com/dhowden/tag.readTextWithDescrFrame(0xc0000da038, 0x3, 0x3, 0x101, 0x122b1a0, 0x0, 0x0)
        /Users/jaylin/go/pkg/mod/github.com/dhowden/tag@v0.0.0-20200828214007-46e57f75dbfc/id3v2frames.go:460 +0x46e
github.com/dhowden/tag.readID3v2Frames(0x114d680, 0xc0000d8000, 0x17, 0xc0000dc000, 0xc0000d8000, 0x0, 0xb)
        /Users/jaylin/go/pkg/mod/github.com/dhowden/tag@v0.0.0-20200828214007-46e57f75dbfc/id3v2.go:364 +0x90b
github.com/dhowden/tag.ReadID3v2Tags(0x114daa0, 0xc0000d8000, 0x1, 0x0, 0x0, 0x0)
        /Users/jaylin/go/pkg/mod/github.com/dhowden/tag@v0.0.0-20200828214007-46e57f75dbfc/id3v2.go:428 +0xde
github.com/dhowden/tag.ReadFrom(0x114daa0, 0xc0000d8000, 0xc0000d6000, 0x17, 0x217, 0x0)
        /Users/jaylin/go/pkg/mod/github.com/dhowden/tag@v0.0.0-20200828214007-46e57f75dbfc/tag.go:52 +0x324
main.main()
        /Users/jaylin/GolandProjects/gofuzz_test/main.go:20 +0xb5
```

dhowden commented on Nov 19, 2020 Owner

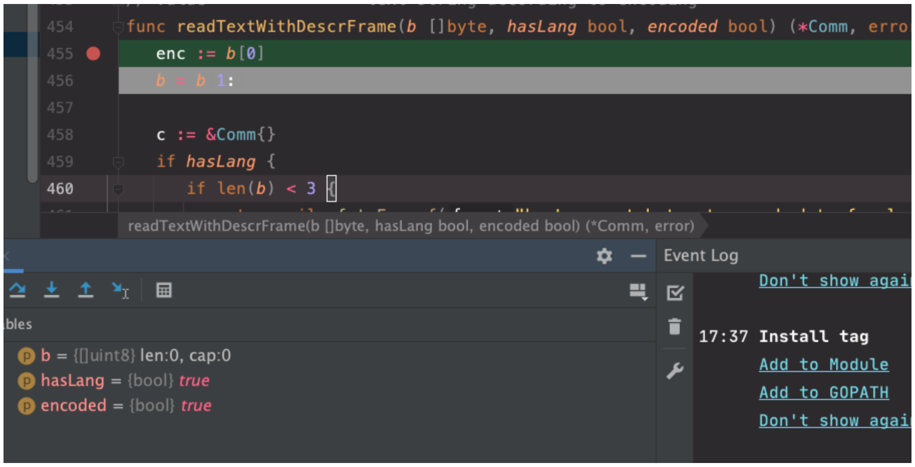
Duplicate of #76

dhowden marked this as a duplicate of #76 on Nov 19, 2020

dhowden closed this as completed on Nov 19, 2020

Jayl1n commented on Nov 19, 2020 Author

If the array size is less than 2, it still panic in the latest commit, just like in the figure below

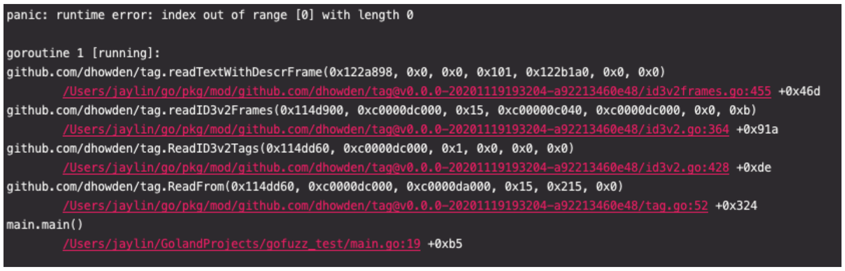


```
func readTextWithDescrFrame(b []byte, hasLang bool, encoded bool) (*Comm, erro
    enc := b[0]
    b = b[1:]

    c := &Comm{}
    if hasLang {
        if len(b) < 3 {
```

readTextWithDescrFrame(b []byte, hasLang bool, encoded bool) (*Comm, error)

b = {[]uint8} len:0, cap:0
hasLang = {bool} true
encoded = {bool} true



```
panic: runtime error: index out of range [0] with length 0

goroutine 1 [running]:
github.com/dhowden/tag.readTextWithDescrFrame(0x122a898, 0x0, 0x0, 0x101, 0x122b1a0, 0x0, 0x0)
        /Users/jaylin/go/pkg/mod/github.com/dhowden/tag@v0.0.0-20201119193204-a92213460e48/id3v2frames.go:455 +0x46d
github.com/dhowden/tag.readID3v2Frames(0x114d900, 0xc0000dc000, 0x15, 0xc00000c040, 0xc0000dc000, 0x0, 0xb)
        /Users/jaylin/go/pkg/mod/github.com/dhowden/tag@v0.0.0-20201119193204-a92213460e48/id3v2.go:364 +0x91a
github.com/dhowden/tag.ReadID3v2Tags(0x114dd60, 0xc0000dc000, 0x1, 0x0, 0x0, 0x0)
        /Users/jaylin/go/pkg/mod/github.com/dhowden/tag@v0.0.0-20201119193204-a92213460e48/id3v2.go:428 +0xde
github.com/dhowden/tag.ReadFrom(0x114dd60, 0xc0000dc000, 0xc0000da000, 0x15, 0x215, 0x0)
        /Users/jaylin/go/pkg/mod/github.com/dhowden/tag@v0.0.0-20201119193204-a92213460e48/tag.go:52 +0x324
main.main()
        /Users/jaylin/GolandProjects/gofuzz_test/main.go:19 +0xb5
```

@dhowden

dhowden commented on Nov 20, 2020 Owner

Ah yes! Thanks. Happy to receive a pull request to fix :-).

dhowden reopened this on Nov 20, 2020

Jayl1n commented on Nov 20, 2020 Author

> Ah yes! Thanks. Happy to receive a pull request to fix :-).

@dhowden I'm Sorry. My code is so terrible, but I can give you an advice if you don't have a better way to fix such bugs.

You could use `recover()` function in which caller to regains control of a panicking goroutine. see more detail

dhowden commented on Nov 20, 2020 Owner

> > Ah yes! Thanks. Happy to receive a pull request to fix :-).
>
> **@dhowden** I'm Sorry. My code is so terrible, but I can give you an advice if you don't have a better way to fix such bugs.

No worries :-)

I will have a look now.

dhowden closed this as completed in `d52dcb2` on Nov 20, 2020

dhowden commented on Nov 20, 2020 Owner

Just to note: the library was built to read data from valid files (and making it conform to all the specs was bad enough, so I mostly ignored safety measures to trap bad files).. Using a fuzzer will likely find lots of issues like this!

If people are using this in production environments, would definitely recommend that they wrap all alls to the library with recover (as you suggest above) to make sure that a panic here does not bring down their entire process.

👍 1 ❤️ 1

wader commented on Nov 20, 2020 Contributor

@Jayl1n **@dhowden** nice work! would be possible to add your fuzzing test code to the repo?

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants