New issue Jump to bottom

## Many panics/crashes when fuzzing #31



rc-mattschwager opened this issue on Aug 22 · 4 comments

Labels

bug

## rc-mattschwager commented on Aug 22

Hi there,

I've been fuzzing this library using the excellent go-fuzz fuzzer. It has produced quite a few panics in the Unmarshal functionality. These panics can have security implications and can lead to, for example, denial-of-service. Ideally this library would be resilient to potentially malicious msgpack payloads.

I used the following code to test Unmarshal against fuzzed inputs:

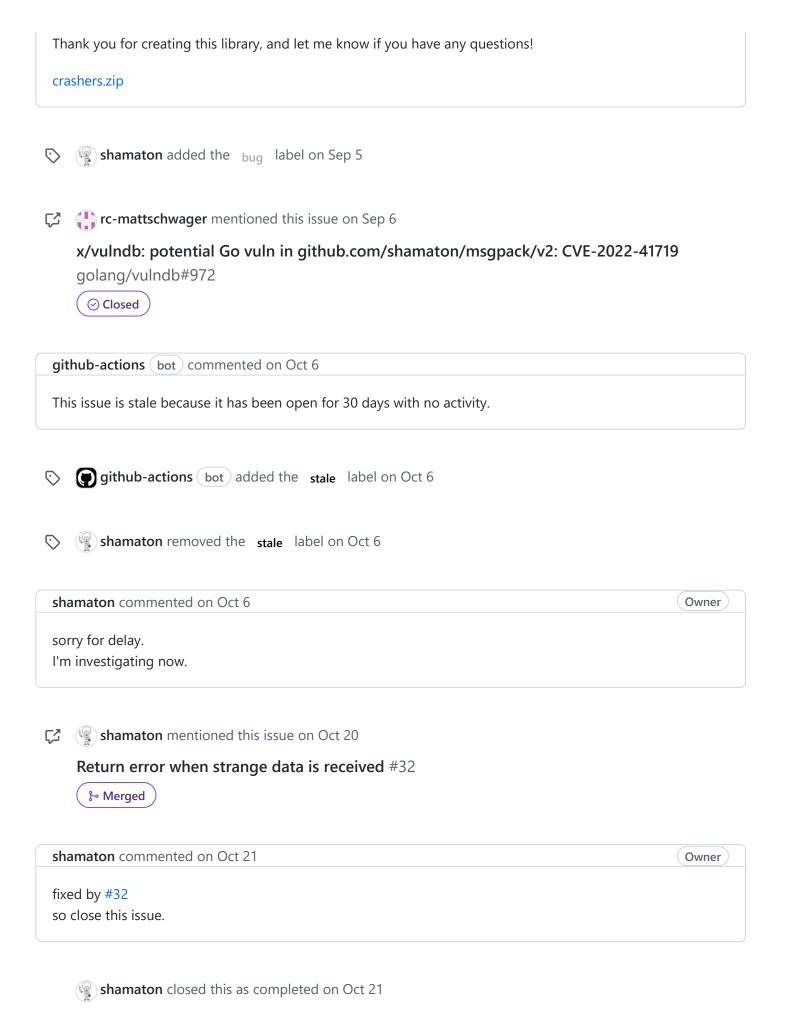
```
var r interface{}
err = msgpack.Unmarshal(data, &r)
if err != nil {
        panic(err)
}
```

This produced a number of crashes. I see three general types of crashes:

```
1:panic: runtime error: slice bounds out of range [:6] with capacity 1
1:panic: runtime error: index out of range [23] with length 23
1:panic: runtime error: hash of unhashable type map[interface {}]interface {}
```

There are many more like this, but they're the same crash with different values.

I've attached a zipfile with files that contain msgpack data that produce the panics. Calling Unmarshal on the data like the above code snippet should reproduce the crashes. Due to potential security implications here, it would be beneficial if this msgpack implementation was resilient to these payloads, and produced errors on invalid input instead of panics. Is it possible to fix the code such that it is?



Nice! Thank you for looking int	o this!		
<u>1</u>			
Assignees			
No one assigned			
Labels			
bug			
Projects			
None yet			
Milestone			
No milestone			
Development			
No branches or pull requests			
2 participants			

Author

rc-mattschwager commented on Oct 25