Go Booby Traps 4

Michel Casabianca
casa@sweetohm.net

Go programming language is easy to learn, but there are some tricky traps. This article series is trying to show these booby traps so that you avoid them.

Let's say we have `User` structures and we want to make a list of names. As there may be a lot of them, we would like to build a slice of pointers to these names. We could write following code (on the Playground) :

```
package main

type User struct {
        Name string
}

func getNames(users []User) []*string {
        var names []*string
        for _, user := range users {
                names = append(names, &user.Name)
        }
        return names
}

func main() {
        users := []User{{Name: "foo"}, {Name: "bar"}}
        names := getNames(users)
        for _, name := range names {
                println(*name)
        }
}
```

But when we run this code, we get:

```
$ go run broken.go
bar
bar
```

Why is this code broken? How would you fix it?

Explanation

When we iterate on `users` with `range`, Go reuses the same variable for `user` and thus it is at the same memory address. So at each iteration, the address of `Name` field of `user` is identical and pointer has the same value. Thus the name in the whole slice points to the `Name` value in the last iteration.

To fix this, we can create a variable for the name as follows (on the Playground) :

```
package main

type User struct {
        Name string
}

func getNames(users []User) []*string {
        var names []*string
        for _, user := range users {
                name := user.Name
                names = append(names, &name)
        }
        return names
}

func main() {
        users := []User{{Name: "foo"}, {Name: "bar"}}
        names := getNames(users)
        for _, name := range names {
                println(*name)
        }
}
```

Variable `name` is created at each iteration, thus `Name` field is at a different memory address and code works as expected.

Conclusion

This example is taken from a real bug :

- Issue description: https://github.com/long2ice/swagin/issues/6
- Diff for the fix: https://github.com/long2ice/swagin/pull/7/files

Enjoy!