Denial-of-Service (SIGSEGV) at xmlquery.(*Node).InnerText #39

Closed **dwisiswant0** opened this issue on Aug 29, 2020 · 0 comments

Labels enhancement

dwisiswant0 commented on Aug 29, 2020 • edited

Summary

The `LoadURL` function allows all response types/formats to be parsed *(other than XML)*, so that it can proceed to the next process (e.g. `xmlquery.(*Node).InnerText` from `xmlquery.FindOne`) without validation.

Description

This security issue affects all `xmlquery` version.

Steps to Reproduce

```go
package main

import (
        "fmt"
        "github.com/antchfx/xmlquery"
)

func main() {
        wadl, err := xmlquery.LoadURL("https://httpbin.org/get")
        if err != nil {
                panic(err)
        }

        attr := xmlquery.FindOne(wadl, "//application/@xmlns")
        fmt.Println(attr.InnerText())
}
```

The logs will look similar to the following:

```
panic: runtime error: invalid memory address or nil pointer dereference
[signal SIGSEGV: segmentation violation code=0x1 addr=0x28 pc=0x6a179a]

goroutine 1 [running]:
github.com/antchfx/xmlquery.(*Node).InnerText.func1(0xc00032e2d0, 0x0)
        /home/dw1/.go/src/github.com/antchfx/xmlquery/node.go:55 +0x2a
github.com/antchfx/xmlquery.(*Node).InnerText(0x0, 0x746b67, 0x14)
        /home/dw1/.go/src/github.com/antchfx/xmlquery/node.go:67 +0x84
main.main()
        /tmp/xmlquery.go:15 +0xa0
exit status 2
```

Vulnerable code:

xmlquery/node.go
Lines 50 to 62 in 64ca73d

```go
50                      switch n.Type {
51                      case TextNode, CharDataNode:
52                              buf.WriteString(n.Data)
53                      case CommentNode:
54                      default:
55                              for child := n.FirstChild; child != nil; child = child.NextSibling {
56                                      output(buf, child)
57                              }
58                      }
59              }
60
61              var buf bytes.Buffer
```

Recommended Mitigations

- Validates the response from URLs loaded in `LoadURL` , if not XML format; then returns an error.
- Using `xml.Unmarshal` *(only to check the validity of the XML content).*

zhengchun added the **enhancement** label on Aug 30, 2020

zhengchun added a commit that referenced this issue on Aug 30, 2020

`checking XML formatted from HTTP response #39` ✕ 5648b2f

dwisiswant0 closed this as completed on Sep 16, 2020

quetzyg mentioned this issue on Sep 20, 2020

[FIX] Parsing XML from a URL #41

Merged

2 tasks

lamados mentioned this issue on Jan 28, 2021

Out-of-date version of xmlquery introduces security vulnerability gocolly/colly#581

Closed

dwisiswant0 mentioned this issue on Aug 4, 2021

Denial of Service (DoS) in antchfx/xmlquery dwisiswant0/advisory#1

Closed

Assignees

No one assigned

Labels

enhancement

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants