New issue

Jump to bottom

Path Traversal Attacks #1226

Closed **egovorukhin** opened this issue on Feb 21 · 6 comments

Labels bug security

egovorukhin commented on Feb 21

Hello, I found a problem when requesting - path traversal attacks (https://localhost/..%5clogs/app.log). If you specify a backslash (%5c) character in the path, then you can follow the path /../ and get data from the root. It may be worth adding a check for part of the path - /... strSlashDotDotBackSlash = []byte("/..."). At your discretion. Thanks.

erikdubbelboer added a commit that referenced this issue on Feb 23

`Warn about unsafe ServeFile usage` ... ✓ `8086be4`

erikdubbelboer mentioned this issue on Feb 23

Warn about unsafe ServeFile usage #1228

Merged

erikdubbelboer commented on Feb 23 Collaborator

I'm guessing this is about the unsafe usage of one of the ServeFile functions? They allow anything as they use `/` as root for all the requests. Using user specified paths for these functions is very unsafe, I had added some documentation changes for that here: #1228

Or was this about another issue?

egovorukhin commented on Feb 24 Author

I wrote an example for demonstration https://github.com/egovorukhin/pathTraversalAttacks, using fiber(https://github.com/gofiber/fiber). Checking for the correctness of the path in the function fasthttp->uri.go->normalizePath(dst, src []byte) []byte.

erikdubbelboer commented on Feb 24 Collaborator

I'm not seeing any issues with your example repo:

```
% git clone git@github.com:egovorukhin/pathTraversalAttacks.git
% go mod vendor
% go run main.go &
% curl 'http://localhost:3003/..%5csecret.txt'
Cannot GET /..%5csecret.txt
% curl 'http://localhost:3003/..%5c..%5clogs/secret.txt'
Cannot GET /..%5c..%5clogs/secret.txt
% curl 'http://localhost:3003/..%5cclogs/secret.txt'
Cannot GET /..%5cclogs/secret.txt
% curl 'http://localhost:3003/home'
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <h1>Hello world!</h1>
</body>
</html>
```

Are you maybe on a Windows machine? Does running these commands result in something different for you?

egovorukhin commented on Feb 24 Author

Yes, app run on a Windows Cluster. Could you add a fix for such cases?! Please.

erikdubbelboer commented on Feb 24 Collaborator

`fasthttp.FS` is completely incompatible with Windows. See #1108 and #1101.

Now that you have shown that it's also not secure on Windows I'm wondering if I should prevent the use of `fasthttp.FS` on windows by throwing an error. Either that or someone needs to take the time to make `fasthttp.FS` compatible with windows.

 **erikdubbelboer** added bug security labels on Feb 24

egovorukhin commented on Feb 24 Author

I added the code to the normalizePath function and it solved my problem

file **strings.go**

```
var (
   ...
  strSlashDotDotBackSlash = []byte(`/..\`)
  strBackSlashDotDotBackSlash = []byte(`\..\`)
  ...
)
```

file **uri.go**

```
func normalizePath(dst, src []byte) []byte {
...

// remove /foo/..\ parts
        for {
                n := bytes.Index(b, strSlashDotDotBackSlash)
                if n < 0 {
                        break
                }
                nn := bytes.LastIndexByte(b[:n], '/')
                if nn < 0 {
                        nn = 0
                }
                n += len(strSlashDotDotBackSlash) - 1
                copy(b[nn:], b[n:])
                b = b[:len(b)-n+nn]
        }

        // remove /foo\..\ parts
        for {
                n := bytes.Index(b, strBackSlashDotDotBackSlash)
                if n < 0 {
                        break
                }
                nn := bytes.LastIndexByte(b[:n], '/')
                if nn < 0 {
                        nn = 0
                }
                n += len(strBackSlashDotDotBackSlash) - 1
                copy(b[nn:], b[n:])
                b = b[:len(b)-n+nn]
        }

...
}
```

 **erikdubbelboer** closed this as completed in `6b5bc7b` on Feb 28

---

erikdubbelboer added a commit that referenced this issue on Mar 3

 `Warn about unsafe ServeFile usage (#1228)` ··· ✕ `15262ec`

 **ruokeqx** mentioned this issue on Sep 4

fix: fix path-traversal bug cloudwego/hertz#229

Merged

2 tasks

Assignees

No one assigned

Labels

bug **security**

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants