go-macaron / macaron Public

New issue

Jump to bottom

# vulnerability: open redirect in static handler #198

Closed ev0A opened this issue on Apr 30, 2020 · 6 comments · Fixed by #199

| | |
|---|---|
| Assignees | |
| Labels | bug |

**ev0A** commented on Apr 30, 2020 • edited

看了这个issue，觉得蛮有意思，试了一下发现这个问题在另一种场景里可以发生
#125
需要开启静态文件服务
漏洞代码

```
package main
import "gopkg.in/macaron.v1"
func main() {
        m := macaron.Classic()
        m.Get("/", func(ctx *macaron.Context) string {
                return "Hello world!"
        })
        m.Use(macaron.Static("static"))
        m.Run()
}
```

只要使用了 m.Use(macaron.Static("static"))静态文件服务，可以在所有浏览器下触发任意302跳转漏洞
比如访问： `http://127.0.0.1:4000//evoa.me%2f..`
即可跳到evoa.me

我也好奇地看了一下源码，大概逻辑是当访问的路由不在注册的路由中时，就会去查找访问的路径否是在注册的静态目录中
分析一下，以 `http://127.0.0.1:4000//evoa.me%2f..` 为例
macaron.v1/static.go:121

静态资源的判断逻辑是staticHandler函数实现的，126行会获取到访问的路径，值为 `//evoa.me/..` ，138行会使用http.FileSystem去打开访问路径，由于我们访问的路径是..结尾，所以http.FileSystem返回是一个文件夹对象，150行fi.IsDir()会为真，进入if判断，152行，我们的路由后缀并不是'/'，继续进入if判断，153行给我们访问的路径添加一个'/'后缀，然后返回response，所以http 返回头对应的location为

`Location: //evoa.me/../`
//开头的host，浏览器默认会用当前协议去重定向，即可造成任意302跳转问题

至于之前issue提的chrome无法成功，具体原因就是chrome会自动把访问路径的/..给标准化（即删除），从而导致/..无法发送至后端，只要将/.. 编码为 %2f..即可逃逸chrome的标准化
至于302的危害，如果是钓鱼的话确实不是很大，但是302漏洞除了钓鱼以外最主要的是可以和其他场景或漏洞结合，产生更大的危害。
举个例子
比如假如程序里有这么一个代码

```
package main

import (
        "gopkg.in/macaron.v1"
        "io/ioutil"
        "net/http"
)
func httpGet(url string) string {
        resp, err := http.Get(url)
        if err != nil {
                // handle error
        }
        defer resp.Body.Close()
        body, err := ioutil.ReadAll(resp.Body)
        if err != nil {
                // handle error
        }
        return string(body)
}
func main() {
        m := macaron.Classic()
        m.Get("/:username", func(ctx *macaron.Context) string {
                return ctx.Params(":username")
        })
        m.Get("/get",func(ctx *macaron.Context) string {
                return httpGet("http://127.0.0.1:4000/"+ctx.Query("username"))
        })
        m.Use(macaron.Static("static"))
        m.Run()
}
```

可能实际代码不会这么写，但是如果是RPC调用，或者后端要和其他web进行交互，这种场景就很常见了。
这个代码本身并没有任何问题，但是如果存在一个302跳转，恶意攻击者可以输入
`http://127.0.0.1:4000/get?username=/192.168.10.1/..`
由于http.Get默认会进行302
即可造成SSRF（服务端请求伪造漏洞），攻击者可访问所有内网web内容

by the way, django曾经也有一个差不多的漏洞，CVE-2018-14574 可以看看这篇文章：https://xz.aliyun.com/t/3302
修复建议：
对重定向的url进行判断，不允许//开头

由衷的感谢您看完这么长一篇issue报告

**humaidq** commented on Apr 30, 2020 Contributor

I am not able to reproduce this vulnerability. I get a 404 with the
code provided and my own applications I previously developed.

I am using Go 1.14.2 with latest release of Macaron.

**ev0A** commented on Apr 30, 2020 Author

> I am not able to reproduce this vulnerability. I get a 404 with the code provided and my own applications I previously developed. I am using Go 1.14.2 with latest release of Macaron.

Did U make the directory named "static" ?

*This comment has been minimized.* Sign in to view

**humaidq** commented on Apr 30, 2020 Contributor

Nevermind, can reproduce when I put the dots .. in the end. My bad.

**humaidq** mentioned this issue on Apr 30, 2020

**static: clean the path URL before redirecting** #199

Merged

**ev0A** changed the title ~~vulnerablilitiy: 302重定向漏洞~~ **vulnerability: 302重定向漏洞** on Apr 30, 2020

**unknwon** added the bug label on May 2, 2020

**unknwon** commented on May 2, 2020 Member

Thanks for the feedback, impressive!

**unknwon** changed the title ~~vulnerability: 302重定向漏洞~~ **vulnerability: open redirect in static handler** on May 2, 2020

**unknwon** assigned **humaidq** on May 3, 2020

**unknwon** closed this as completed in #199 on May 3, 2020

**unknwon** commented on May 3, 2020 Member

New release is tagged: https://github.com/go-macaron/macaron/releases/tag/v1.3.7

Thanks **@humaidq** for fixing this!

❤️ 2

**simonpasquier** mentioned this issue on Jun 26, 2020

**Upgrade gopkg.in/macaron.v1 to v1.3.7 (or above)** grafana/grafana#25856

Closed

**ruokeqx** mentioned this issue on Sep 4

**vulnerability: open redirect in static handler** labstack/echo#2259

Closed

Assignees

humaidq

Labels

bug

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

static: clean the path URL before redirecting

3 participants