Vendor of the products: Tenda

Affected products: Tenda TX3 V16.03.13.11_multi

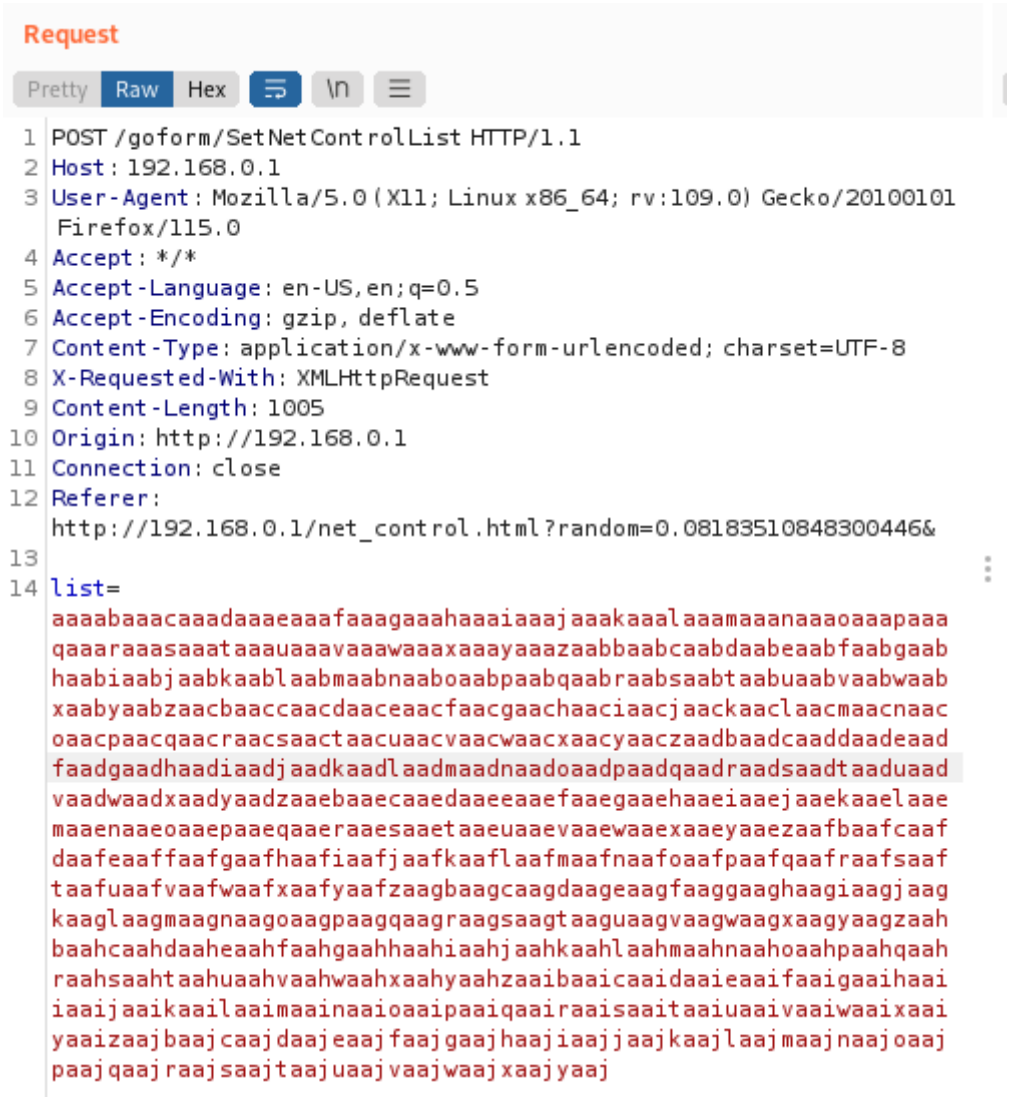
Hardware Link: https://www.tendacn.com/tw/download/detail-4015.html

Vulnerability Description

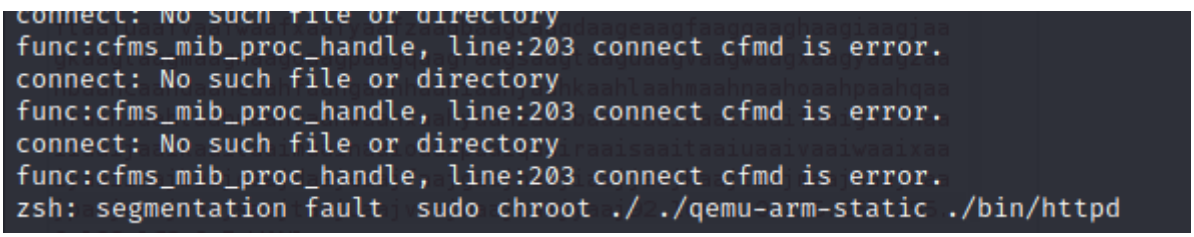
A buffer overflow vulnerability was discovered in Tenda TX3 V16.03.13.11_multi, triggered by the list parameter at /goform/SetNetControlList. This vulnerability allows attackers to cause a Denial of Service (DoS) via a crafted packet.

POC

send



You can see that the router has crashed.



Code in httpd

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

```
websFormDefine((char_t *)"getPptpOnlineClient", formgetPptpOnlineClient);
websFormDefine((char_t *)"setPptpUserList", formSetPPTPUserList);
websFormDefine((char_t *)"SetPptpClientCfg", formSetPPTPClient);
websFormDefine((char_t *)"GetPptpClientCfg", formGetPPTPClient);
websFormDefine((char_t *)"GetVpnStatus", formGetVpnStatus);
websFormDefine((char_t *)"GetAdvanceStatus", formGetAdvanceStatus);
websFormDefine((char_t *)"SetNetControlList", formSetQosBand);
websFormDefine((char_t *)"GetNetControlList", formGetQosBand);
formDefineUcloudv2();
websFormDefine((char_t *)"notNowUpgrade", formNotNowUpgrade);
websFormDefine((char_t *)"getHomeLink", formGetHomeLink);
websFormDefine((char_t *)"GetLEDCfg", formGetSchedLed);
websFormDefine((char_t *)"SetLEDCfg", formSetSchedLed);
websFormDefine((char_t *)"getMacFilterCfg", formGetMacFilterCfg);
websFormDefine((char_t *)"setMacFilterCfg", formSetMacFilterCfg);
websFormDefine((char_t *)"SetFirewallCfg", formSetFirewallCfg);
websFormDefine((char_t *)"GetFirewallCfg", formGetFirewallCfg);
websFormDefine((char_t *)"SetIPTVCfg", formSetIptv);
websFormDefine((char_t *)"GetIPTVCfg", formGetIptv);
websFormDefine((char_t *)"GetStaticRouteCfg", formGetRouteStatic);
websFormDefine((char_t *)"SetStaticRouteCfg", fromSetRouteStatic);
websFormDefine((char_t *)"SetDDNSCfg", formSetSysToolDDNS);
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.

```
void __fastcall formSetQosBand(webs_t wp, char_t *path, char_t *query)
{
  unsigned __int8 *v4; // r4
  int v5; // r0
  int v6; // r2
  unsigned __int8 cgi_debug[16]; // [sp+10h] [bp-190h] BYREF
  unsigned __int8 ret_buf[32]; // [sp+20h] [bp-180h] BYREF
  unsigned __int8 guest_down_speed[32]; // [sp+40h] [bp-160h] BYREF
  unsigned __int8 guest_up_speed[32]; // [sp+60h] [bp-140h] BYREF
  unsigned __int8 msg_info[256]; // [sp+80h] [bp-120h] BYREF

  memset(ret_buf, 0, sizeof(ret_buf));
  memset(msg_info, 0, sizeof(msg_info));
  v4 = websGetVar(wp, (char_t *)"list", (char_t *)&byte_7A45B);
  unSetQosOldMiblist();
  set_qosoldMib_list();
  unSetQosMiblist();
  set_qosMib_list(v4, 0xAu);
  memset(guest_down_speed, 0, sizeof(guest_down_speed));
  memset(guest_up_speed, 0, sizeof(guest_up_speed));
  GetValue("wl.guest.down_speed", guest_down_speed);
  memset(cgi_debug, 0, sizeof(cgi_debug));
  if ( GetValue("cgi_debug", cgi_debug) && !strcmp("on", (const char *)cgi_debug) )
    printf(
      "%s[%s:%s:%d] %s%s == %s\n\x1B[0m",
      "\x1B[0;33m",
      "cgi",
      (const char *)_func___13078,
      2073,
      "\x1B[0;32m",
      "wl.guest.down_speed",
      (const char *)guest_down_speed);
  strcpy((char *)guest_up_speed, (const char *)guest_down_speed);
  set_wl_guest_qos_list(guest_up_speed, guest_down_speed);
  if ( CommitCfm(v5) )
  {
```

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);
```