Talos Vulnerability Report

TALOS-2021-1388

# Sealevel Systems, Inc. SeaConnect 370W MQTTS Certificate Validation vulnerability

FEBRUARY 1, 2022

### CVE NUMBER

CVE-2021-21959

### Summary

A misconfiguration exists in the MQTTS functionality of Sealevel Systems, Inc. SeaConnect 370W v1.3.34. This misconfiguration significantly simplifies a man-in-the-middle attack, which directly leads to control of device functionality.

### Tested Versions

Sealevel Systems, Inc. SeaConnect 370W v1.3.34

### Product URLs

SeaConnect 370W - https://www.sealevel.com/product/370w-a-wifi-to-form-c-relays-digital-inputs-a-d-inputs-and-1-wire-bus-seaconnect-multifunction-io-edge-module-powered-by-seacloud/

### CVSSv3 Score

7.7 - CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:H/A:H

### CWE

CWE-295 - Improper Certificate Validation

### Details

The SeaConnect 370W is a Wi-Fi connected IIoT device offering programmable cloud access and control of digital and analog I/O and a 1-wire bus.

This device offers remote control via several means including MQTT, Modbus TCP and a manufacturer-specific protocol named “SeaMAX API.”

The device is built on top of the TI CC3200 MCU with built-in Wi-Fi capabilities.

The SeaConnect 370W communicates with the remote “Sealevel SeaCloud” via MQTTS communications. While establishing this connection, the device chooses to explicitly ignore certificate validation errors, making man-in-the-middle type attacks substantially simpler to conduct. The initialization of the MQTTS connection occurs in the function `GetConnected`, located at raw offset 0x10446 which is mapped to RAM for execution at 0x20010446. The function responsible for creating a secure socket is titled `NetworkConnectTLS`, located in RAM at 0x2001BEB4 and referenced within `GetConnected` at offset 0x7A. The prototype of the `NetworkConnectTLS` function is approximately `int NetworkConnectTLS(Network *n, char *host, int port, void *certificate_filename, int certificate_filename_len)` and when no `certificate_filename` is supplied the socket will not perform any validation of the supplied server certificate.

Below, we see the portion of `GetConnected` where the MQTTS client is established.

```
NewNetwork(network);
MQTTClientInit(mqtt_client, network, 10000, *p_tx_buffer, 512, *p_rx_buffer, 512);
mqtt_client->defaultMessageHandler = DefaultCallback;
mqtt_conf.port = 8883;
rc = NetworkConnectTLS(network, mqtt_conf.host, 8883, 0, 0);
```

Observe that `NetworkConnectTLS` is called with no `certificate_filename` provided and therefore no certificate validation is conducted.

Successfully performing a man-in-the-middle of the MQTTS connection would allow an unauthenticated remote attacker to make modifications to the state of the outputs, manipulate the reported states of inputs and even convince the device to apply a malicious firmware update.

### Timeline

2021-10-21 - Initial vendor contact
2021-10-26 - Vendor disclosure
2022-02-01 - Public Release

### CREDIT

Discovered by Francesco Benvenuto and Matt Wiseman of Cisco Talos.