SYSE 569, Cybersecurity for Systems Engineers, Spring 2025

PROJECT Elements based on Portions

Assigned: 21 January, 2025

Due: 6 May, 2025

Name: Spencer Beer

OPTION B: Project by Ethical Hacking

A Survey of a Wireless Gateway for Marine Electronics

CONTENTS

BACKGROUND

The Gateway is a thumb-sized bridge that plugs straight into an NMEA 2000 backbone and sends a boat's data over 2.4 GHz Wi-Fi. It can create its own wireless hotspot or join an onboard router. It converts traffic both ways between NMEA 2000, NMEA 0183, and a RAW protocol stream. It also has the capability to control Raymarine SeaTalk NG autopilot computers. The Gateway can be used for tablet navigation, cable-less NMEA bridges, or data logging.

The manufacturer has reported rapid growth since its founding in 2014. By late 2017, the company noted a 500% increase in sales over 2016. The momentum continued next year – in 2018 they announced they had sold 3× more units than in 2017 and expanded to a network of about 40 dealers worldwide. Even during the pandemic and semiconductor shortages, demand grew: sales in 2021 were 26% higher than 2020 and 44% higher than 2019. Although exact unit counts were not disclosed, these growth rates imply a substantial user base. (If sales were in the low hundreds in 2016, a 5× jump by 2017 and 3× by 2018 would put annual units in the thousands by 2018.) With continued double-digit growth through 2021, it's reasonable to estimate total Wi-Fi gateway devices in circulation in the high thousands (possibly approaching the low tens of thousands) globally by the mid-2020s, though exact figures remain proprietary.

The gateway plugs into either a male DeviceNet Micro-C plug or a female SeaTalk NG spur. Once the device is powered, by default, it hosts its own SSID with a default password. A user then can browse the web page and log in with the default credentials.

But there is one thing that fails to occur: a first-boot password reset. Across consumer, small-business and industrial environments, multiple independent data sources converge on the same conclusion: most users never reset the factory credentials. That's why modern laws (PSTI UK, California SB-327, forthcoming EU RED Cyber requirements) now force manufacturers either to ship a unique per-device secret or to block network access until the very first password change is made [1]. Although the manufacturers mention the need to change the default password in their documentation, it should now be evident that the gateway is not considered secure, and why it was tested in this paper for further vulnerabilities.

Owners of cruising yachts and sailboats have integrated these gateways to feed data into charting apps (OpenCPN, Navionics, Signal K, etc.) and even to cloud services for remote monitoring. By 2019, the manufacturer introduced a free Cloud service to upload voyage tracks and live telemetry when the device is internet-connected. This indicates that some fraction of the user base connects their gateways to marina Wi-Fi or cellular routers to enable remote access. The company's year-end reports also mention OEM projects and use by charter/fleet companies, implying these devices are not only DIY aftermarket gadgets but are finding their way into professional deployments on commercial or rental vessels as well.

Protocols

Marine applications use a variety of protocols to communicate between each other, whether that's for safety, navigation, or environmental monitoring. These protocols encompass both wireless and physical networking systems, each serving specific functions to ensure efficient and secure maritime operations.

- CAN Bus (Controller Area Network) – A multi-master serial communication protocol. It was originally developed for automotive applications, and eventually found its way into the foundation of NMEA 2000 [2]
- NMEA 2000 (IEC 61162-3) – A plug-and-play communications standard used in marine sensors, and display units within ships and boats. It typically operates at 250 kbps and is based on the J1939 standard [3]
- NMEA 0183 – An earlier standard that uses a single talker, multiple listener configuration over a 4800 bps data rate [4]
- Raymarine SeaTalk NG – Raymarine's proprietary networking protocol that is electrically compatible with NMEA 2000 but uses different connectors and cabling. It enables integration of Raymarine devices and third-party NMEA 2000-compliant equipment [5]
- Simrad SimNet – Simrad's proprietary network protocol derived from NMEA 2000 [6]
- Automatic Identification System (AIS) – Transmits vessel information such as identification, position, course, and speed to other ships and stations. Its main purpose is to provide situational awareness and collision avoidance [7]

There are other wireless communications standards available for the maritime industry; however, the scope of this project is specific to the features mentioned on the gateway's user manual.

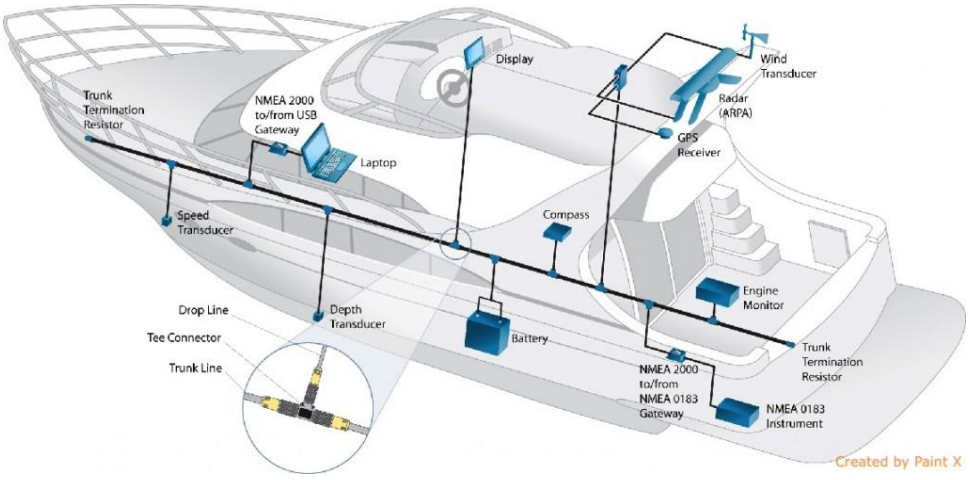


Figure 1, Representation of a Marine Networking

GOALS

The primary objective of this research is to understand what the security model of aftermarket marine IoT electronics looks like. Since the automotive industry is growing in technology, so are the ways it should be secured. Internet of Things (IoT) devices provide convenience and integration but often lack robust cybersecurity practices. This paper evaluates whether marine-grade devices – such as the gateway – inherit the same oversight issues observed in other consumer IoT domains, particularly in how they handle authentication, update mechanisms, encryption, and network exposure.

An example of such research is the Electronic Logging Device (ELD) in the Medium-Heavy Duty (MHD) automotive domain [8]. Although these devices differ in the sense that they are mandated, and the gateway is not, they are still important in many boaters' electronic systems. After all, they allow us access to physical networks required to completely control a marine system (i.e., NMEA 0183, 2000 protocols). The manufacturer mentions, "You can share your boat's data to registered users of the Cloud service or to anyone by a "secret" link. A few minutes after your boat's network connects to the Internet, the data will be in the Cloud and your loved ones will be able to know that everything is fine with you". Although the cloud service for this paper is considered out of scope, the topic of connectivity between these IoT enabled devices is becoming increasingly important.

Figure 2, Representation of Threat Scope

The manufacturer promotes the system as a reliable tool for accident investigations and regatta data analysis. However, the device is highly susceptible to Man-in-the-Middle (MitM) attacks, allowing an adversary to easily intercept, monitor, and inject malicious messages in either direction. More importantly, the user remains unaware of the intrusion, which can result in falsified data, potentially undermining its credibility in legal proceedings. Ultimately, this research aims to help the marine industry recognize and communicate the critical importance of securing IoT devices. Within a limited timeframe, the study focuses on identifying the full range of vulnerabilities present in the gateway.

INSTALLATION AND RUNNING

For this project, the Kali Linux operating system serves multiple different software utilities supported by the open-source hacking community [9]. In it, a few stand out and are used for their different purposes in the research:

- Nmap – used to reveal what ports and services are running on a host [10]
- Wireshark – used to sniff live network traffic as it comes across [11]
- Burp Suite – used to intercept browser traffic, tamper with requests, and fuzz parameters [12]
- Binwalk – used to analyze the data of binary blobs, and extract relevant details [13]
- Strings – used to quickly sift through common patterns of Unicode and ASCII text in firmware [14]
- Ghidra – used to reverse engineer and decompile executables [15]
- Airmon-ng – allows a network interface to capture all wireless traffic in range, not just traffic addressed to it [16]
- Airodump-ng – collects information such as SSIDs, BSSIDs, encryption types, channel numbers, and connected clients, and saves packets for later analysis or cracking [17]

And some separate freeware used in the firmware analysis:

- Esptool – used for interacting with the ESP8285 chip available on the YDWG-02 [18].
- Esp2elf – used to convert an ESP8266 Read-Only Memory (ROM) into an Executable and Linkable Format (ELF) file for Ghidra [19]
- Logic pro 2 – used to analyze digital and analog signals in the reverse engineering process [20]
- KRACK (Key Reinstallation Attack) – Affects WPA2's 4-way handshake, which allows attackers within range to manipulate and replay cryptographic handshakes, forcing reinstallation of already-in-use keys. This results in nonce, reuse, breaking encryption, and enabling packet decryption, injection, and hijacking [21]

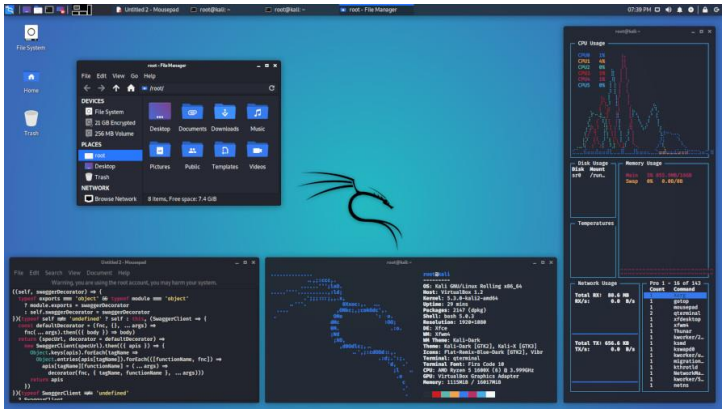


Figure 3, Desktop Screenshot of Kali Linux [9]

Execution Workflow

Good reverse engineering always starts with understanding the hardware used in the product. After taking the product apart, two images were taken from the top and bottom sides of the wireless gateway, seen in Figures 4 and 5. There are two chips visible on the top side of the printed circuit board (PCB) shown in figure 1. The FCC ID on the top of the metal enclosure of the SoC tells us what is inside without performing a chip-off maneuver, and the user manual from the Docters of Intelligence & Technology Co. LTD. It is an ESP8285 SoC, typically used in IoT applications. In this case, there are two different targets for exploration. The first target, the STM32, was difficult to solder 30 AWG wire too (see Figure 6). As shown in figure 4, the ESP-M2 already has solder pads that make firmware extraction a little easier.

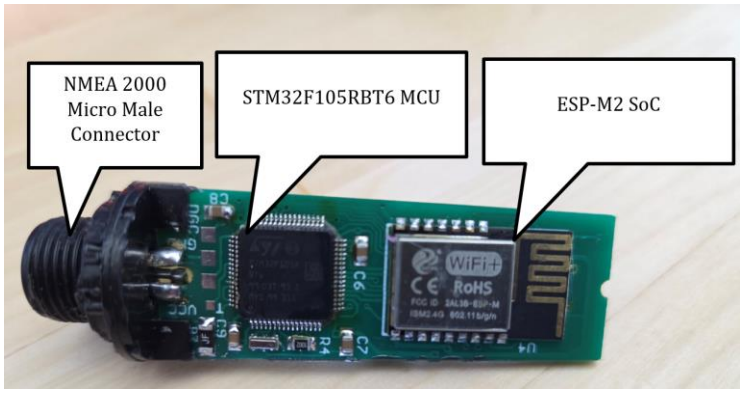


Figure 4, "Top Side" of the Gateway

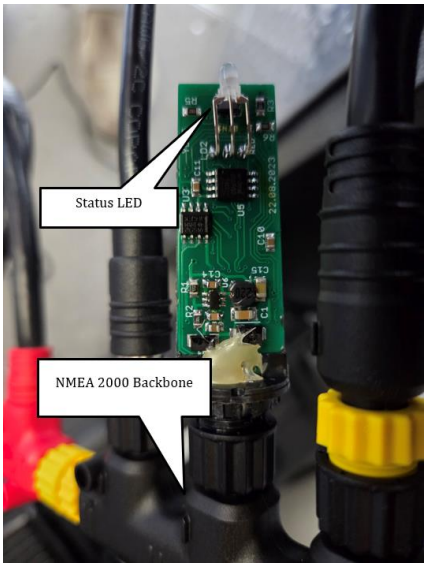
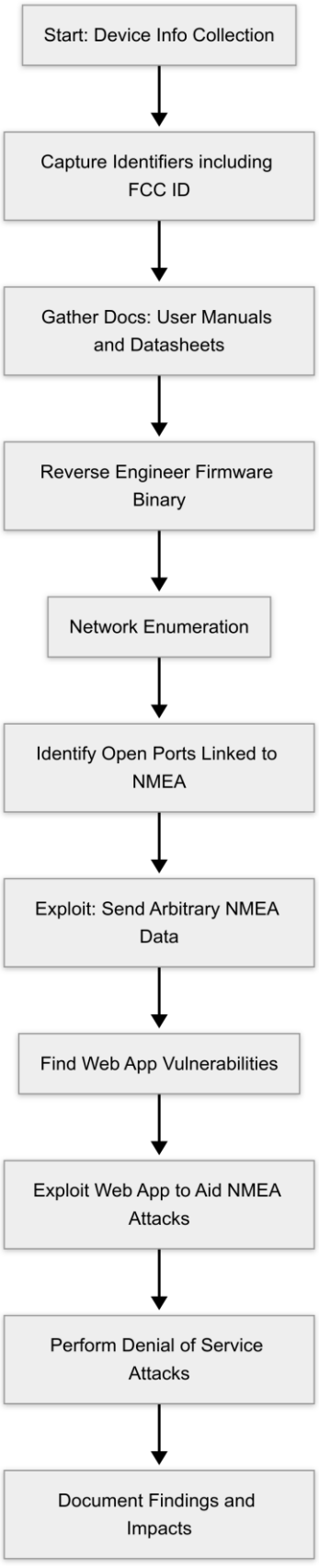


Figure 5, "Bottom Side" of the Gateway



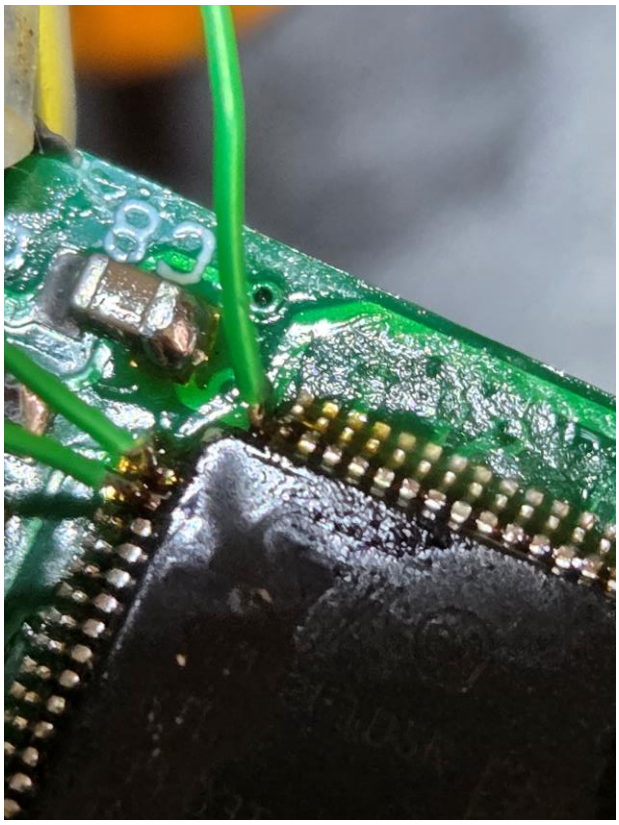


Figure 6, 30 AWG Wire Representation in Respect to STM32

It's important to note that the wireless update mechanism occurs over-the-air (OTA), over an unencrypted http channel. While this isn't secure, the binary was found to be encrypted at the least, before performing any further hardware modifications (see Figure 7).

```
binwalk -E WUPDATE.BIN

DECIMAL     HEXADECIMAL    ENTROPY
--------------------------------------------------------------------------------
0           0x0            Rising entropy edge (0.972447)
120832      0x1D800        Rising entropy edge (0.977437)
```

Figure 7, Entropy Analysis with Binwalk

The binary "WUPDATE.BIN" is publicly accessible, but it's also important to note that the binary can be captured and reassembled after capturing the update with Wireshark.

Performing a bit of reverse engineering on the ESP-M2 chip gives more information on the device. First, we connect to the pins shown in Table 1, then extract the chip contents. The bootloader also tells us important information as well, shown in Figure 8.

ESP8285 Pin	Serial Adapter
VCC (3.3v)	3.3V
GND	GND
TXD	RX
RXD	TX
GPIO0	GND
EN (RST)	3.3V (pulled up)

Table 1, Connections Made for Firmware Extraction with FTDI Cable

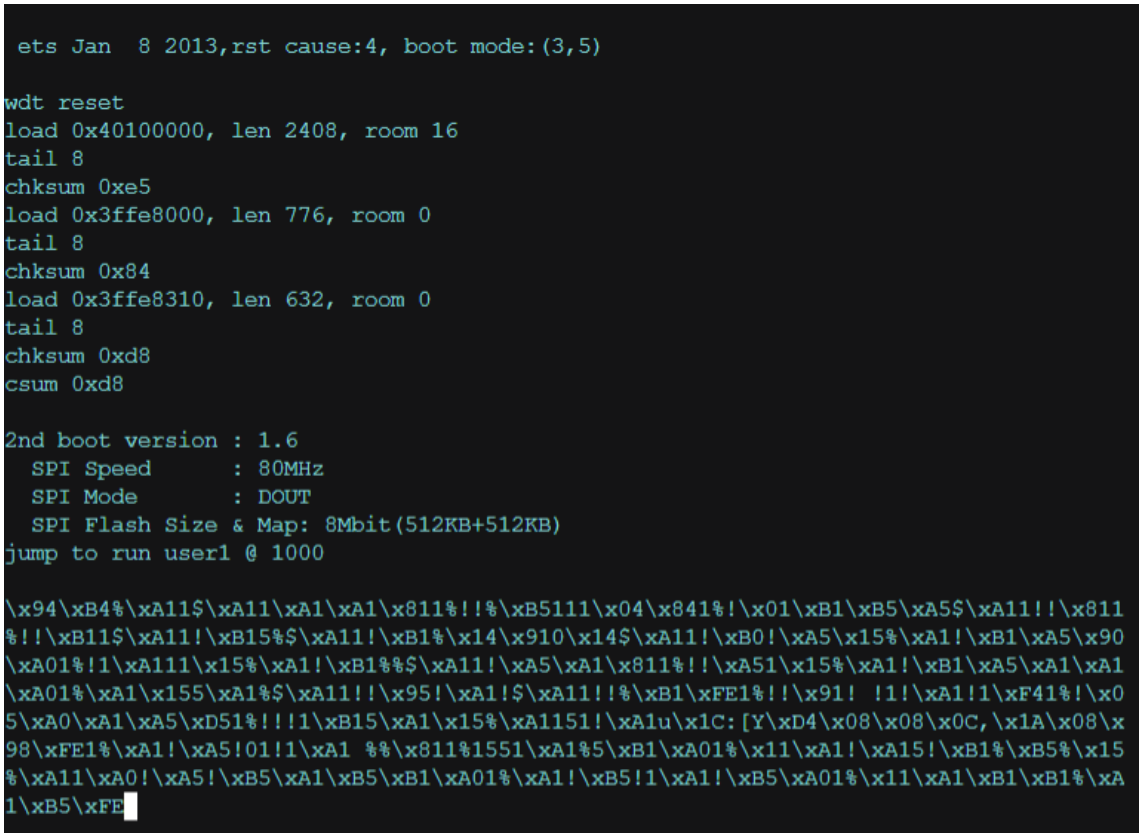


Figure 8, Boot logs of the ESP8285 Chip

After this, the next step is to use the manufacturer's tool to dump the firmware from the chip (esptool). The following command was used to extract the contents of the chip: `*esptool.py -p PORT -b 115200 read_flash 0 ALL flash_contents.bin*` [https://docs.espressif.com/projects/esptool/en/latest/esp8266/esptool/basic-commands.html]. Another quick scan with esptool gives us exact image information:

```
esptool image_info flash_contents.bin

esptool.py v4.8.1
File size: 2097152 (bytes)
Detected image type: ESP8266
Image version: 1
Entry point: 40100438
3 segments

Segment 1: len 0x00968 load 0x40100000 file_offs 0x00000008 [IRAM]
Segment 2: len 0x00308 load 0x3ffe8000 file_offs 0x00000978 [DRAM]
Segment 3: len 0x00278 load 0x3ffe8310 file_offs 0x00000c88 [DRAM]
Checksum: d8 (valid)
```

Figure 9, ESP8285 YDWG-02 Firmware Image Info

Things like offsets can help use load and identify sections in Ghidra. Thankfully, a tool was written to automate this process and allow Ghidra to automatically assembly the XTENSA binary as best as possible: *`esp2elf flash_contents.bin flash_contents.elf`* [19]. It was also found that the device can send commands from the ESP8285 to the STM32 chip to the NMEA 2000 network after dumping strings on the firmware as well. Some of the commands include:

```
NULL
SYNC
WIFI_STATUS
ADD_CB
NMEA_GETSERVERS
NMEA_SEND
STM_LOG
STM_SET_TIME
STM_REQ_WIFIMODE
STM_GET_SERIAL
FILTER_GETFILTERS
CMD_STM_REQ_PING
CMD_STM_GET_STATISTICS
CMD_XDR_GET_RESP
CMD_FLASH_RESP
CMD_LOGGING_RESP
CMD_YACHTD_CONNECT
CMD_YACHTD_DATA
CMD_YD_AUTOPILOT
```

Figure 10, Binary String Dump of ESP8285 Firmware

Given the timeframe of this research, and the interesting commands found in Figure 10, the next step was immediately to evaluate the security of the web application, and the gateway, apart from lower-level binary analysis.

The next step was reconnaissance. After 13.2 seconds, we get information regarding the http server mentioned in manufacturer's user manual:

```
nmap.exe --top-ports 1000 -Pn 192.168.4.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-24 14:17 Mountain Daylight Time
Nmap scan report for 192.168.4.1
Host is up (0.0054s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 4E:EB:D6:DE:83:3A (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 13.20 seconds
```

Figure 11, Nmap on Windows 11 – Initial Enumeration from Within the Network

```
Nmap scan report for 192.168.10.71
Host is up (0.0063s latency).
Not shown: 65526 closed tcp ports (reset)
PORT     STATE    SERVICE
80/tcp   open     http
490/tcp  filtered micom-pfs
1456/tcp open     dca
1458/tcp open     nrcabq-lm
20178/tcp filtered unknown
25715/tcp filtered unknown
36412/tcp filtered unknown
54978/tcp filtered unknown
61439/tcp filtered netprowler-manager
MAC Address: 4C:EB:D6:86:BA:CC (Espressif)
```

Figure 12, Nmap on Windows 11 – Deeper Enumeration from Within the Network

This is where things get serious. There are a few ports mentioned in the user manual of the gateway that suggest data can be sent or received. The next step was to evaluate these ports. Immediately, it is recognized that port 1456/tcp is dumping unencrypted and unauthenticated NMEA 0183 data, also mentioned as default in the user manual (see Figure 13).

```
ncat.exe 192.168.10.71 1456
$PCDIN,01F211,003A0605,6F,00475D64070000FF*53
$MXPGN,01F211,686F,00475D64070000FF*1C
$PCDIN,01F211,003A0FD3,6F,00475D64070000FF*51
$MXPGN,01F211,686F,00475D64070000FF*1C
$PCDIN,01F211,003A19A2,6F,00475D64070000FF*2B
$MXPGN,01F211,686F,00475D64070000FF*1C
$PCDIN,01F211,003A2370,6F,00475D64070000FF*56
$MXPGN,01F211,686F,00475D64070000FF*1C
```

Figure 13, Netcat on Windows 11

This is important data because, as mentioned previously, this data can contain things like positioning, gps data, navigation information, wind and weather, heading and compass, and AIS target info. However, it's not completely insecure by default. You cannot send data to the device, unless the user has set the device to bidirectional communication. Still, this is very likely, as the documentation states, "Note that Gateway server port must be configured to work in both directions («Transmit Only» in factory settings) to allow control of autopilot from the application.". If any other servers are turned on, data can be sent or received, based on the configuration by the intended user.

This was tested and shown to allow an attacker complete command and control over the NMEA 2K bus, and potentially, and entire marine system (see Figure 14).

Figure 14, Command and Control of a Multi-Function Display

The gateway also allows for control of autopilots, such as Raymarine. If an attacker sits in the middle, they can potentially spoof waypoints and coordinates for the autopilot, causing it to travel in unintended directions. As mentioned before, these ports are thankfully closed off by default for sending data, but they can be turned on quite easily. The next section goes into detail about lateral movement towards sending data nonetheless, given the device sends data unencrypted.

Lateral Movement and More Exploitation

After some vulnerability scanning with OWASP ZAP and Burp Suite, there were a few vulnerabilities found within the web application of the gateway itself. It's important to note that attackers must be on the same network to perform these attacks.

Attackers can bypass credentials used in the gateway. Cleartext credentials pass through an HTTP request, handled by AJAX, as shown in Figure 15 below.

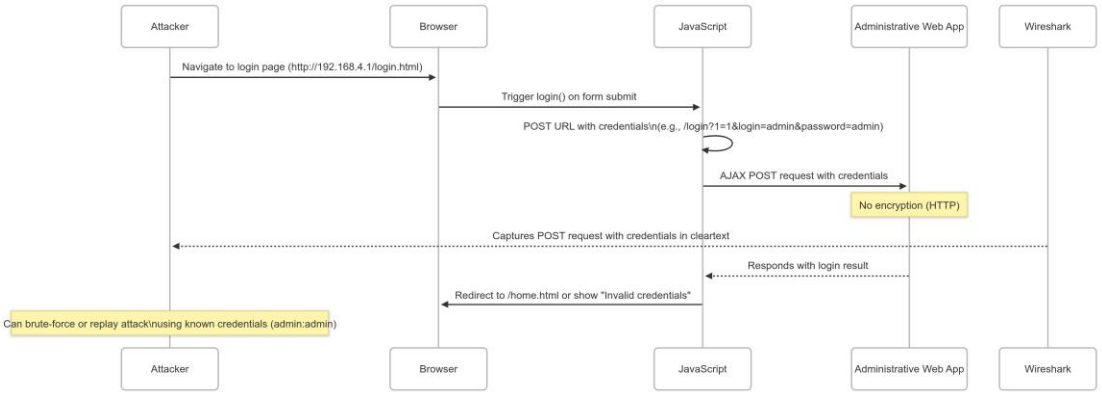


Figure 15, Authentication Bypass Mechanisms

Not only web app credentials, but also credentials of the access point, configurations, hardware info, cloud application info, and security keys. Things like URL handlers used on the website to allow attackers to simply sit and listen for keywords related to their interest in attack vector.

For example, an attacker could be connected to the LAN of the gateway, listening for the `/wifi/connect` endpoint and intercept the packet to connect to their malicious gateway, or even worse, figure out the credentials for a protected network. This example request can be intercepted as shown in Figure 16, below.

```
POST /wifi/connect?essid=<redacted>&passwd=<redacted> HTTP/1.1
Host: 192.168.4.1
Content-Length: 0
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/135.0.0.0 Safari/537.36
Accept: */*
Origin: http://192.168.4.1
Referer: http://192.168.4.1/client.html
Accept-Encoding: gzip, deflate, br
Cookie: session=D033E22AE348AEB5660FC2140AEC35850C4DA997
Connection: keep-alive
```

Figure 16, Hijack POST Request

Figure 16 also shows us the cookie used in the session. The cookies used by the device are not signed and are used to authenticate the user. This means that we can easily inject a cookie into the request, and gain access to the device. This is just another way we can bypass the authentication, since the cookie is a SHA1 hash of the password, and there is no way to change or add users to prevent this. An article from Google's security research team shows how easily these hashes can be broken [22].

Some other vulnerabilities found include the ability for an attacker to perform a DoS on the TCP ports of the gateway. A quick command, shown in Figure 17, prevents the user from using the NMEA features of the gateway.

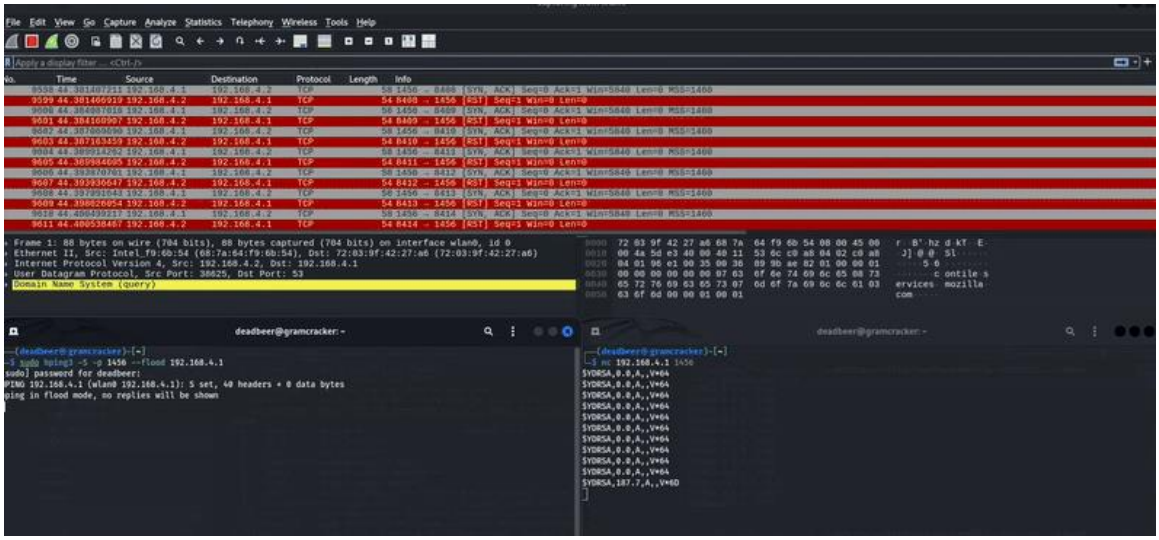


Figure 17, Denial of Service

CODE WRITTEN

Some code written for this project includes scripts to prove the viability of an attacker to move quickly within this scope.

NMEA 0183 Spoofer

An NMEA 0183 spoofer python script, developed to assist in the process of sending remote frames to the target gateway and port:

```
python3 ./nmea0183spoofer.py --help
usage: nmea0183spoofer.py [-h] --type {serial,udp,tcp} --target TARGET --sentence
SENTENCE [--baud BAUD]

Spoof an NMEA 0183 sentence. This program also comes with quick references of NMEA
sentences if you run it with --docs.

options:
  -h, --help          show this help message and exit
  --type {serial,udp,tcp}
                      Output type: serial, udp, or tcp
  --target TARGET     Serial device (/dev/ttyUSB0) or IP:PORT
  --sentence SENTENCE   NMEA sentence body (without $ or *checksum)
  --baud BAUD         Baud rate for serial (default: 4800)
```

Figure 18, NMEA 0183 Spoofer Usage

NMEA 2K Spoofer

An NMEA 2000 spoofer python script, developed to assist in the process of sending remote RAW frames to the target gateway and port:

```
python3 ./nmea2kspoofer.py --help
usage: nmea2kspoofer.py [-h] [--ip IP] [--port PORT] [--proto {tcp,udp}] [--file] [--docs] [log]

Replay or spoof CAN logs to YDWG-02 in RAW format.

positional arguments:
  log             String of log formatted messages OR path to a log file.

options:
  -h, --help        show this help message and exit
  --ip IP           Target IP address (default: 192.168.4.1)
  --port PORT       Target port (default: 1456)
  --proto {tcp,udp}  Protocol (default: tcp)
  --file            Treat log as a file path, not raw string
  --docs            Display usage examples and common NMEA 2000 PGNs
```

Figure 19, NMEA 2000 Spoofer Usage

Yacht Destroy

This script was created to check for the AP the static IP address, or the user can manually enter the IP address of the device. If the device is in AP mode, it tries to connect to the AP with the default password, otherwise it performs a de-authentication attack and tries to crack the hash. Once the script has entered the same network as the device, it scans for ports. If port 80 is active, it tries to authenticate with the web application using the default credentials. If any other ports are found open (given a supplied range to check via Nmap), the script then listens for valid NMEA 2K, or 0183 traffic (these can be bidirectionally set up; hence they are targeted). Then, if the default credentials do not work, it sets up a listener on the Linux device to capture the traffic. If it detects passwords, or other key words, it will alert the user. Example usage is shown below in figure x.

```
./yachtDestroy --deauth
./yachtDestroy --deny
./yachtDestroy --inject
./yachtDestroy --enumerate
```

Figure 20, Yacht Destroy Usage

RESULTS

The vulnerabilities found in this device include the following, based on CVSS score ranges:

Category	CWE Reference	Severity
Hardcoded Credentials	CWE 798	High
Cookie Injection & Weak Hashing	CWE 565/ 326	High
Cleartext Transmission	CWE 319 / 311	Critical
Unauthenticated Function Access	CWE 306 / 285	High
XSS / Clickjacking	CWE 79 / 1021	Medium
Protocol Spoofing	CWE 345	High
TCP Flood	CWE 400	Medium
Exposed Cloud Endpoint Key	CWE 200 / 639	High
Broken Access Control	CWE 639	High

Table 2, Marine Gateway Vulnerabilities

These vulnerabilities were verified through live interaction with the device in both Access Point and client modes. Spoofed messages were successfully injected into the NMEA 2000 bus. Authentication could be bypassed using cookie injection, and session hijacking was feasible due to SHA-1 hash reuse. TCP ports, including the main NMEA 0183 endpoint (port 1456), were unencrypted, unauthenticated, and susceptible to SYN flood denial-of-service attacks. Several web application vulnerabilities including unauthenticated reboot, XSS, and IDOR were also identified.

DISCUSSION

Prior studies into automotive and industrial IoT devices have revealed similar issues: insufficient authentication, weak cryptography, and poor protocol hardening. This project confirms that marine electronics suffer from many of the same architectural oversights, particularly regarding default credentials, lack of encryption, and user-unaware attack surfaces. While automotive devices are increasingly regulated, marine IoT remains loosely governed by security mandates, with few mechanisms for enforceable patch management or monitoring.

All testing was conducted on a device owned by the researcher and isolated from public infrastructure. Firmware extraction and network fuzzing were performed within a controlled lab environment. No external services were accessed without authorization. Any disclosure of findings to the vendor will follow the industry-standard responsible disclosure timelines and mechanisms.

Reverse engineering and firmware analysis proved critical in exposing the depth of vulnerability present. The availability of community-supported tools like Ghidra, Esptool, and Burp Suite significantly accelerated the research. Additionally, this project reinforced the reality that user misconfiguration (such as enabling bidirectional NMEA or not changing default passwords) can amplify flaws. Access control is imperative in the modern world, especially with IoT. The gateway here provides proof.

RECOMMENDATIONS AND CONCLUSIONS

To mitigate the risks discovered during this project, several vendor-side remediations are recommended. First, the device should enforce a first-use password reset by disabling all network functions until a unique password is set. All communications should be secured with TLS, replacing unencrypted HTTP for both web access and firmware updates. API endpoints must require session validation for both web and TCP interfaces to prevent unauthorized access. NMEA protocol streams, especially those configured for bidirectional use, should be encrypted using TLS or a lightweight alternative to prevent spoofing. Session management must be hardened by replacing insecure SHA-1-based cookies with signed, time-limited, and scoped tokens. To reduce the risk of brute force and denial-of-service attacks, per-IP rate limiting should be implemented. Public cloud URLs should be made non-indexable to prevent unauthorized data exposure via search engines. Finally, firmware updates should include signature verification and secure boot mechanisms to ensure the integrity and authenticity of all OTA-delivered software.

Future Work

Further research could explore Cloud-based exploitation and indirect object reference attacks on scale. More research into the firmware and cryptanalysis of any proprietary communication schemes between the ESP8285 and STM32 should prove more findings.

Conclusion

This project demonstrates that the gateway, like many embedded IoT devices, suffers from critical security oversight ranging from plaintext credentials to unauthenticated network injection. These vulnerabilities pose substantial risks not only to vessel safety and privacy but also to the credibility of the data used in high-stakes environments like regattas or investigations. Addressing these issues will require stronger default configurations, vendor cooperation, and user awareness. The broader maritime industry must begin to treat cybersecurity as an integral part of onboard system reliability.

REFERENCES

Note: Lots of the references that could be included in this paper are redacted for security purposes.

[1] D. Bonderud, "Router reality check: 86% of default passwords have never been changed," *IBM Think Blog*, Jan. 3, 2025. [Online]. Available: https://www.ibm.com/think/insights/router-reality-check-86-percent-default-passwords-have-never-been-changed.

[2] Maretron, "IBEX 2005 Presentation," presented at the International Boatbuilders' Exhibition and Conference (IBEX), 2005. [Online]. Available: https://www.maretron.com/company/pubs/IBEX%202005%20Presentation.pdf.

[3] SAE International, "SAE J1939 Standards Collection on the Web," [Online]. Available: https://www.sae.org/standards/development/ground-vehicle/sae-j1939-standards-collection-on-the-web.

[4] Wikipedia contributors, "NMEA 0183," *Wikipedia*, The Free Encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/NMEA_0183.

[5] Raymarine, "SeaTalk NG and NMEA 2000," *Raymarine*, [Online]. Available: https://www.raymarine.com/en-us/our-products/networking-and-accessories/seatalk-ng-and-nmea-2000.

[6] Simrad AS, *Simrad SimNet Installation Manual*, Rev. A, Nov. 15, 2004. [Online]. Available: https://busse-yachtshop.de/pdf/Simrad-SimNet-Installation-Manual-bys.pdf. [Accessed: May 8, 2025].

[7] International Telecommunication Union, "Radiocommunications for keeping ships and people safe at sea," *ITU News*, Sep. 28, 2023. [Online]. Available: https://www.itu.int/en/mediacentre/backgrounders/Pages/Radiocommunications-for-keeping-ships-and-people-safe-at-sea.aspx.

[8] J. Jepson, R. Chatterjee, and J. Daily, "Commercial Vehicle Electronic Logging Device Security: Unmasking the Risk of Truck-to-Truck Cyber Worms," in *Proc. 2nd Symp. Vehicle Security and Privacy (VehicleSec 2024)*, San Diego, CA, USA, Feb. 2024, pp. 1–12. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/vehiclesec2024-47-paper.pdf.

[9] Offensive Security, "Kali Linux Tools," *Kali Linux*, [Online]. Available: https://www.kali.org/tools/.

[10] G. Lyon, "Nmap: the Network Mapper," [Online]. Available: https://nmap.org/

[11] Wireshark Foundation, "Wireshark: Go Deep," [Online]. Available: https://www.wireshark.org/.

[12] PortSwigger Ltd., "Intercepting HTTP traffic with Burp Proxy," [Online]. Available: https://portswigger.net/burp/documentation/desktop/getting-started/intercepting-http-traffic.

[13] ReFirm Labs, "Binwalk: Firmware Analysis Tool," GitHub, [Online]. Available: https://github.com/ReFirmLabs/binwalk.

[14] Dfreshalot, "Firmware Analysis Part 1," *Dfreshalot Blog*, Jan. 31, 2021. [Online]. Available: https://dfresh.ninja/index.php/2021/01/31/firmware-analysis-part-1/.

[15] National Security Agency, "Ghidra," [Online]. Available: https://ghidra-sre.org/.

[16] Aircrack-ng, "Airmon-ng," [Online]. Available: https://www.aircrack-ng.org/doku.php?id=airmon-ng.

[17] Aircrack-ng, "Airodump-ng," [Online]. Available: https://www.aircrack-ng.org/doku.php?id=airodump-ng.

[18] Espressif Systems, "esptool: Serial utility for flashing, provisioning," GitHub, [Online]. Available: https://github.com/espressif/esptool.

[19] R. Burton, "esp2elf: Convert esp8266 ROM to ELF file," GitHub, [Online]. Available: https://github.com/raburton/esp2elf.

[20] Saleae Inc., "Logic 2 Software," [Online]. Available: https://www.saleae.com/downloads/.

[21] M. Vanhoef and F. Piessens, "Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2," *KRACK Attacks*, [Online]. Available: https://www.krackattacks.com/.

[22] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full SHA-1," *SHAttered.io*, Feb. 23, 2017. [Online]. Available: https://shattered.io/static/shattered.pdf.