Demo:
Persistent Firmware-Level Compromise in a Maritime Autopilot System

Carson Green
Colorado State University

Rik Chatterjee
Colorado State University

Jeremy Daily
Colorado State University

Abstract

This work presents a demonstration of arbitrary Controller Area Network (CAN) message injection on a maritime National Marine Electronics Association (NMEA) 2000 network via firmware compromise. By reverse engineering a firmware update binary for a marine autopilot computer, we identify and modify low-level CAN transmission routines to inject spoofed messages, including rudder control commands and address claims. The attack exploits the absence of authentication and cryptographic integrity checks in the firmware update mechanism. An adversary with access to a chart plotter can deliver a tampered update via an SD card, causing the autopilot to accept and install the malicious firmware. Upon reboot, the compromised autopilot executes attacker-controlled code, enabling persistent and arbitrary CAN message injection. This work highlights systemic security deficiencies in embedded maritime systems and demonstrates the risks posed by unauthenticated firmware distribution in safety-critical navigation infrastructure. To the best of our knowledge, this is the first publicly demonstrated instance of firmware-based CAN injection in a maritime context.

1 Introduction and Background

Modern vessels ranging from recreational boats to commercial ships and unmanned surface vehicles are increasingly reliant on embedded networked systems for navigation, propulsion, and situational awareness [1]. Among these, autopilot computers play a critical role in maintaining course and interfacing with GPS receivers, heading sensors, and rudder actuators [2]. These subsystems typically communicate over the NMEA 2000 protocol [3], a marine-specific implementation of the Controller Area Network (CAN) designed for plug-and-play interoperability and simplified wiring.

Despite its widespread adoption, NMEA 2000 lacks essential security features such as message encryption, authentication, or integrity checks [4]. As this protocol is increasingly deployed in commercial platforms, these weaknesses pose growing risks to the safety and reliability of maritime systems.

This work targets one such system, a marine autopilot computer that accepts unauthenticated firmware updates via the NMEA 2000 network. The following section outlines our methodology for identifying and exploiting this weakness to achieve arbitrary CAN message injection through firmware compromise.

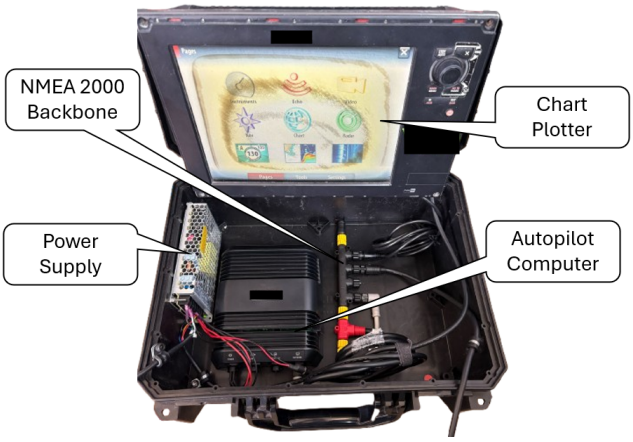

Figure 1: Autopilot Testbed with Connected Chart Plotter

2 Attack Methodology

Our evaluation setup, shown in Fig. 1, consists of a chart plotter connected to a marine autopilot computer via the NMEA 2000 network, mirroring a typical vessel deployment. We began by accessing the autopilot unit and identifying an exposed, unlocked JTAG interface connected to an NXP LPC2368 microcontroller. Using this interface, we extracted the onboard firmware for reverse engineering and analysis.

Through static analysis, we located the CAN transmission subroutines responsible for sending messages on the NMEA 2000 network. These routines were modified to introduce attacker-controlled payloads, including spoofed rudder control commands and address claims. To deploy the modified

firmware, we obtained a publicly available firmware update package in the vendor's proprietary '.swup' format. These '.swup' files are XML-encoded containers that include the binary firmware image encoded in Motorola S-Record format. After disassembling the container, we replaced the original S-Record contents with our modified binary, recalculated the S-Record checksums, and repackaged the update file to preserve the original structure. The final attack phase uses a chart plotter to deliver the tampered update. The '.swup' file is loaded onto an SD card and inserted into the chart plotter, which transfers the update over the NMEA 2000 network as illustrated in Fig. 2. Critically, the firmware update process lacks any authentication, signature validation, or integrity checks. As a result, the autopilot accepts and installs the malicious firmware without resistance. Upon reboot, the modified firmware executes the injected CAN subroutines, enabling persistent, arbitrary message injection on the network as shown in Fig. 3.

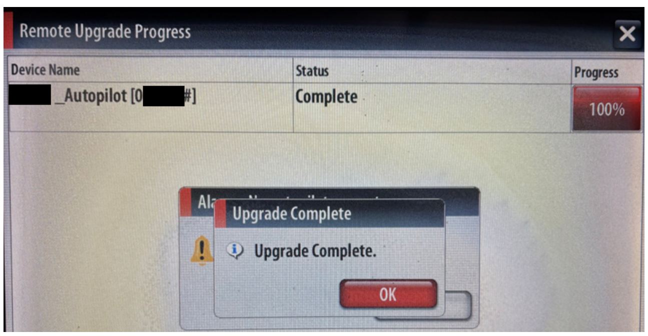

Figure 2: Chart Plotter Showing Completion of Malicious Firmware Update

```
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
can0  19FA0302   [8]  8F FF FF 7F FF 7F FF 7F
can0  18EEFF00   [8]  00 00 00 00 00 00 00 00
```

Figure 3: Arbitrary CAN Message Injection from compromised Autopilot

3 Demonstration

The demonstration presents the attack described in the previous section, performed on a live hardware setup consisting of a chart plotter and a marine autopilot connected via the NMEA 2000 network as shown in Fig. 1. We show the autopilot receiving a malicious firmware update from the chart plotter using an SD card loaded with a tampered '.swup' file. Once the update is initiated, viewers can observe the firmware being accepted and installed by the autopilot without authentication. After reboot, the device begins injecting attacker-controlled CAN messages onto the network. Injected traffic includes address claim messages, which are captured using a CAN monitoring interface and visualized live. The demonstration highlights the complete attack chain from firmware tampering to persistent message injection.

4 Conclusion and Security Implications

This work demonstrates how trusted update channels such as removable media and onboard chart plotters can be leveraged to introduce persistent compromise into maritime control systems. Even in air-gapped or isolated deployments, the lack of firmware authentication exposes critical components to tampering. The attack model echoes prior compromises in industrial systems, underscoring the urgent need for secure firmware validation in safety-critical maritime environments.

5 Responsible Disclosure

At the time of writing, the authors would like to note that this vulnerability has been reported to the vendor, as well as to MITRE for consideration for a Common Vulnerabilities and Exposures (CVE) ID.

References

[1] National Marine Electronics Association. NMEA 2000 Vulnerability to Cyberattacks and Mitigation. https://digitalyacht.net/wp-content/uploads/2024/10/NMEA-2000-Vulnerability-to-Cyberattacks-and-Mitigation2024.pdf, 2024.

[2] Larry Anderson. Leading the Way in NMEA 2000. https://www.maretron.com/company/pubs/IBEX%202005%20Presentation.pdf, 2005.

[3] NMEA 2000 Appendices A & B – Parameter Groups (PGNs) NMEA Network Messages. Standard NMEA 2000, National Marine Electronics Association, Severna Park, MD, 2015. https://www.nmea.org/nmea-2000.html.

[4] Gary C. Kessler. The CAN Bus in the Maritime Environment – Technical Overview and Cybersecurity Vulnerabilities. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, 15(3):531–540, 2021. https://doi.org/10.12716/1001.15.03.05.