Demo: ELDSentry: A Simple DIY J1939 Firewall to Prevent CAN Bus Attacks via ELDs or the Diagnostic Port

Joseph Olatt, Samuel Hollifield, Joel Asiamah, Rima Aswar Awad
Oak Ridge National Laboratory

Abstract

ELDSentry is a simple do-it-yourself (DIY) microcontroller-based firewall that prevents injection of arbitrary CAN messages onto the CAN bus via devices connected to the SAE J1939 diagnostic port. Additionally, the system logs CAN data to a microSD card. The primary objective was to prevent CAN attacks via the wireless interfaces (WiFi, Bluetooth Low Energy, etc.) found on most ELDs that have been mandated by the Federal Motor Carrier Safety Administration (FMCSA) since 2016 for most commercial motor vehicles (CMV).

The setup uses common off-the-shelf components, like the Teensy microcontroller, CAN Transceivers, DC voltage regulators, etc. The objective is to make the hardware bill of materials, wiring diagram and software code freely accessible, with the vision/hope that CMV operators can procure the hardware portion of this device from any entity that is independent of someone like an ELD vendor, inspect the software that runs it and protect themselves from ELDs (and other similar devices) from being used as a wireless attack surface.

1 Introduction

At the 2024 Symposium on Vehicles Security and Privacy (VehicleSec), researchers at Colorado State University presented a paper on how they used the WiFi access point (AP) of an ELD, installed in a truck, to flood the Controller Area Network (CAN) bus with malicious CAN messages and slow down the truck [1]. They pulled up beside the truck in a separate vehicle and performed this attack wirelessly while the truck was in motion.

Theoretically, a wireless attack like this can also be performed via the Bluetooth interfaces that most ELDs provide to facilitate transfer of electronic log data to authorized safety officials.

The above mentioned attack and reports of numerous other CAN bus attacks have highlighted the ease of carrying out such attacks via the SAE J1939 diagnostic port. When wireless access to a CAN-connected device like an ELD is available, such CAN bus attacks become easier to carry out and could lead to widespread disruptions in commercial fleets, with severe safety and operational implications. Some additional links to articles and videos are provided in the reference section [2, 3].

The vision/hope of this research project is that by making the hardware design and software open source, sharing the knowledge and testing results, will: (1) Give CMV operators a choice in vendors they can procure required hardware from; (2) Offer lowest possible cost due to marketplace competition; (3) Provide ability to audit the firmware running on the device and hence prevent introduction of malware, back doors, etc. in an ELD-type device; (4) Independence from manufacturers of devices that connect to the diagnostic port like ELD manufacturers, etc., providing a separation of tasks/duties, which is an important concept in the world of security; (5) Mitigate CMV cybersecurity vulnerabilities.

A low-cost device like ELDSentry running auditable software will allow any CMV operator to procure and use any of the certified, to FMCSA, ELD device and use it without worrying about cybersecurity attacks via the ELD device of their choice.

This work is still in progress and the prototype is yet to be subjected to field or extensive testing.

2 Hardware Design and Bill of Materials

The hardware design is actually quite simple with a goal to keeping the device low cost and do-it-yourself (DIY), if required. The prototype that was built and tested (basic) uses off-the-shelf components. Figure 1 & Figure 2 shows the prototype. One end of the prototype connects to the SAE J1939 diagnostic port, while the other end receives connections from devices like ELD, etc.

The hardware bill of materials is as follows: (1) Teensy 4.1 microcontroller; (2) SAE J1939 9-pin female to open end pigtail cable; (3) SAE J1939 9-pin pass through male/female to open end pigtail cable; (4) DC-DC buck-mode power module; (5) CAN bus transceiver – 2 nos; (6) Full-size breadboard; (7) Perma-proto quarter-sized breadboard PCB; (8) PCB mount

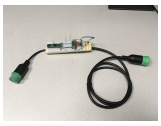

Figure 1: ELDSentry Front View

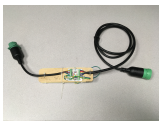

Figure 2: ELDSentry Back View

screw terminal block connectors; (9) 22 AWG wire solid core hookup wires; (10) Micro SD card.

3 Software Design

The firmware code is written in C/C++ utilizing the Teensyduino[1] development environment. The working principle of the software is as follows: The ELDSentry device will pass all traffic it receives from the vehicle CAN bus, on one CAN interface, to the other CAN interface, to which is connected a device like the ELD. All CAN messages that the ELDSentry receives from the vehicle CAN bus are also written to the micro SD card.

All traffic that is generated by a ELD-like device is compared to a white list, and only specific PGNs (Parameter Group Number/Name) are allowed to flow on to the vehicle CAN bus. In the case of request messages (PGN 59904), the CAN message payload can only contain PGNs in the white list. All other traffic is dropped, just like a computer network firewall.

For testing and debugging purposes, the serial port on the Teensy is activated and data written to its serial port can be viewed by connecting to the micro USB port of the Teensy and using a serial communication program like cu(1) or minicom(1).

4 Testing and Preliminary Results

The following test plan was used for testing: (1) Evaluate a limited number of ELDs, identify traffic and requests generated by them and create a white list; (2) Define the white list in the firmware code; (3) Load firmware code onto the ELDSentry prototype; (4) Connect vehicle CAN side of ELDSentry to diagnostic port; (5) Connect the ELD CAN side of the ELDSentry to a SAE J1939 9-pin splitter Y connector; (6) Connect ELD to one end of Y connector and also connect a attack laptop to the other end of Y connector; (7) Use a USB cable to connect a monitoring laptop to the Teensy micro USB port and start up a serial communication program; (8) Turn ignition of vehicle to ON; (9) Verify that ELD has powered up and acquired all the information that it requires (VIN, Engine hours, etc.); (10) Start a fuzzing attack on the attack laptop; (11) The serial communication program on the monitoring laptop should show all attack CAN messages getting dropped and the vehicle operating unperturbed.

[1] https://www.pjrc.com/teensy/teensyduino.html

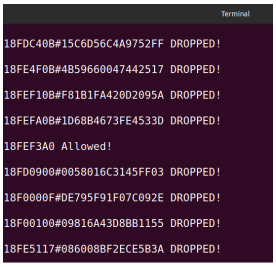

Figure 3: Allowed & dropped messages (Part 1)

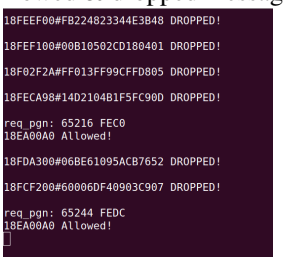

Figure 4: Allowed & dropped messages (Part 2)

Figure 3 and Figure 4 shows screenshots of sample CAN messages allowed and dropped.

5 Acknowledgments

This research work is funded by the Office of Radiological Security (ORS) within the National Nuclear Security Administration under U.S. Department of Energy. The authors also wish to thank the program sponsor at ORS, the ORS Program Management office at ORNL, the Fleet Maintenance Division at ORNL and others that helped in big and small ways.

References

[1] Jeremy Daily Jake Jepson, Rik Chatterjee. Commercial vehicle electronic logging device security: Unmasking the risk of truck-to-truck cyber worm. https://www.ndss-symposium.org/wp-content/uploads/vehiclesec2024-47-paper.pdf, 2024.

[2] Land Line Media. Researchers hack eld, take control of heavy truck. https://www.youtube.com/watch?v=SEON7PMZwDM, 2024.

[3] Josh Rhoten. Researchers highlight potential cybersecurity threats to trucking industry, supply chain. https://engr.source.colostate.edu/researchers-highlight-potential-cybersecurity-threats-to-trucking-industry-supply/, 2024.