Seattle, WA
Sept 11-13, 2025

SECURING LEGACY TRANSPORT PROTOCOLS: REQUIREMENTS-DRIVEN MITIGATION FOR COMMUNICATION SYSTEMS

Adedapo Adeboyejo
Colorado State University
Adedapo.adeboyejo@colostate.edu

Soundarya Sivakumar
Colorado State University
Soundarya.sivakumar@colostate.edu

Spencer Beer
Colorado State University
Spencer.beer@colostate.edu

Jeremy Daily
Colorado State University
Jeremy.daily@colostate.edu

Abstract. Legacy heavy-duty vehicle protocols such as SAE J1708 and J1587 remain in operation due to long vehicle lifespans and the high cost of upgrading diagnostic infrastructures. These standards lack authentication, fairness in bus arbitration, and safeguards for session lifecycle management, leaving them susceptible to protocol-level exploitation. While prior research has examined related vulnerabilities in J1939 and introduced prototype intrusion detection concepts for J1708/J1587, a systematic, requirement-driven evaluation of J1587 vulnerabilities has not yet been performed. This paper investigates protocol-level attacks against J1708/J1587 networks using spoofing and several denial-of-service examples: bus arbitration abuse, spoofing diagnostic traffic, misuse of connection management features. Experimental analysis confirms that these exploitable behaviors are possible and that the protocol itself provides no built-in protection. To address these gaps, we propose requirement-driven mitigations such as limits on concurrent sessions, timeout enforcement for RTS/CTS exchanges, and validation of network messages. These requirements extend the standard and provide a vendor-neutral framework for ensuring predictable and secure diagnostic communication in legacy systems. This study represents the first systematic validation of the legacy trucking protocol and contributes a requirements-based approach for securing outdated but still operational heavy-duty vehicle networks.

Keywords. vulnerabilities, Request to Send (RTS), Clear to Send (CTS), diagnostics, electronic control units (ECUs), legacy protocols, denial-of-service (DoS), spoofing, request overload, mitigation, transport protocol (TP)

1. Introduction

Medium and heavy-duty vehicles critical to national logistics and public infrastructure are complex systems with interconnected Electronic Control Units (ECUs) that communicate using in-vehicle networks. The Society of Automotive Engineers (SAE) issued J1708/J1587 in 1988 as the first mainstream diagnostic standard for heavy-duty vehicles. Through the Open System Interconnection (OSI) model's layered architecture, J1708 operates at the Physical Layer and Datalink Layer, defining RS-485 serial communications and bus arbitration mechanisms [14]. J1587 functions at the Application Layer, while defining a transport protocol, specifying diagnostic message sequencing and multi-frame transfer coordination [11, 12].

SAE J1939, first issued in 1994 and built upon the Controller Area Network (CAN), is also organized in layers like the OSI model's layered architecture [9, 14]. The underlying CAN bus operates at the Physical Layer, providing a more reliable communication system with more integrated arbitration than J1708's RS-485 implementation. J1939 then provides functions at the OSI Datalink, Network, Transport, and Application layers to standardize how ECUs exchange data over CAN by defining message structures, priority handling, and transport mechanisms [6]. The complete J1939 frame format is documented in SAE J1939-21, which also specified multi-frame transfers with the Transport Protocol Connection Management (TP.CM) and Transport Protocol Data Transfer (TP.DT) [6].

The underlying communication networks of heavy-duty vehicles, especially the older ones like SAE J1708 and J1587, may have inherent standard and protocol vulnerabilities that warrant examination. Although J1939 vulnerabilities have received significant attention in prior research, J1587 has not been studied with the same rigor [4, 7]. This is a critical gap because J1587 remains in operation across mixed fleets, aftermarket tools, and older trucks, where vendor-specific implementations vary and resilience is inconsistent. Without standardized safeguards, ECUs are left with unpredictable levels of protection, and security success depends on undocumented Original Equipment Manufacturer (OEM) design choices.

The standards are meant to serve as a guide toward accepted practice and support OEMs in their product development. However, the advent of protocol vulnerabilities detailed in previous research highlights a need for an improvement in these systems. Systems engineering is a guide to the engineering of complex systems and does this through the definition and testing of requirements. This paper highlights the importance of utilizing test cases to verify if provided requirements are satisfied and utilizing this process to propose new requirements.

The overarching goal of this paper is to contribute to the field of systems engineering by demonstrating a case study of utilizing systems engineering requirement principles as they relate to analysis of J1708/J1587 and proposing the results as a roadmap to developing better communication systems in the heavy-truck industry. Through empirical testing, we demonstrate the results of data link layer spoofing, and adoptions of the common Denial-of-Service (DoS) attack. These results did not support our hypothesis that a vendor specific implementation of the standard will not satisfy all the guidance provided. However, the data supported gaps in the standard, which underscores the need for systems engineering requirement-driven mitigations in communication systems.

The remainder of this paper is structured as follows. Section 2 provides a literature review, and a background on systems engineering requirements as they relate to this paper. Section 3 explains this paper's hypothesis and research questions along with the current requirements. Section 4 presents the experimental verification of test cases for the provided requirements. Section 5 explains the results of the test cases along with a discussion of their implications for new proposed requirements. Section 6 concludes with a discussion of contributions, limitations, and directions for future research.

2. Background

2.1 Lit Review

Prior research has laid the foundation for examining security gaps in heavy-duty vehicle networks. Chatterjee et al. and Mukherjee et al. demonstrated that J1939's transport protocol could be systematically exploited through data transfer abuse, spoofing, and session manipulation, validating these behaviors on a production truck [7, 13]. Their work confirmed that structural vulnerabilities in flow control are not just theoretical but create operational risk. Nnaji et al. [4] shifted this focus to legacy networks, presenting one of the first intrusion detection strategies for J1708/J1587. By modeling expected diagnostic traffic and flagging anomalies such as spoofed requests or resource exhaustion, their Legacy Intrusion Detection System (LIDS) showed that lightweight safeguards could detect protocol abuse in a controlled environment.

Together, these efforts highlighted two key gaps that directly shaped our work. First, while J1939 vulnerabilities have been extensively validated, there has been limited systematic exploration of J1587 despite its continued deployment. Second, detection prototypes like LIDS emphasize reactive defense, but they stop short of defining proactive, requirement-driven safeguards that engineers can embed into specifications and procurement practices. Our paper addresses both needs by conducting an OEM empirical study using the Detroit Diesel Electronic Controls (DDEC) engine controller with J1708/J1587 communications. In doing so, we extend Chatterjee's demonstration of exploit feasibility and Nnaji's recognition of legacy risk into a structured, requirements-focused framework for securing J1708/J1587-based networks.

Figure 1, translated directly from the J1587 standard document, illustrates the J1587 Connection Mode Data Transfer sequence between an originating ECU and a receiving ECU. This process uses RTS/CTS to coordinate segmented data delivery, followed by End of Message (EOM) or Connection Abort. The transport protocol imposes constraints such as a 3825-byte maximum message size and single connection support at any given time [11].

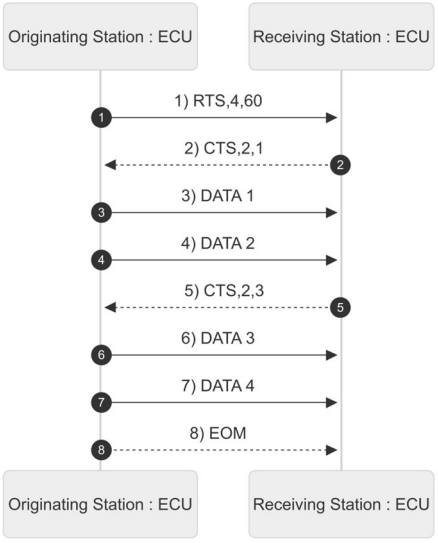

Figure 1. J1587 Connection Mode Data Transfer

Defined message types include:

- **Request to Send (RTS):** Initiates a multi-frame transfer.
- **Clear to Send (CTS):** Responds to RTS, facilitates flow control, and acknowledges received data.
- **End of Message (EOM):** With confirmation of full receipt, this terminates the session.
- **Connection Abort:** Ends the transfer if either ECU cannot continue.

While J1939 provides higher bandwidth and more structured communication than J1587, its transport protocol has some weaknesses. The lack of authentication allows spoofed RTS/CTS or DT messages, while the absence of strict session controls enables request overload and connection exhaustion attacks [3, 7, 13]. These issues may parallel J1587's RTS/CTS vulnerabilities, underscoring that both protocols may share structural flaws at the transport layer despite differences in their underlying physical networks (CAN vs. RS-485) [5].

2.2 Related SAE J1708/J1587 Guidance

The background review includes a deep examination of the J1708 and J1587 standards to extract guidance provided by SAE that establishes requirements for the DDEC implementation to meet. We utilized previous research done on J1939 to highlight areas of interest in the J1708 and J1587 standards. As result, bus access and multi frame transfers are the two main areas of concern in the J1708 message protocol and J1587 standard's transport protocol [1, 2]. In section 4 we create test cases to verify if these guidelines are satisfied by the DDEC engine controller. Below are direct quotes of guidance from SAE J1708 and J1587 that are related to this paper.

Datalink

- Message Identification Character (MID)
    - Guidance 1 - "The first character of every message shall be a MID. The permitted range of MIDs shall include the numbers 0 to 255 [12]."
- Bus Access
    - Guidance 2 - "A transmitter shall begin transmitting a message only after an idle state has continuously existed on the bus for at least a bus access time," where bus access time is defined as, "a time duration equal to the minimum time of an idle line plus the product of 2 bit times and the message priority [12]."

Transport protocol

- Connection Management Functions Guidance
    - Guidance 3 - "Connection mode data will be passed only at the lowest priority of the network; therefore, connection mode data messages may well be interspersed with other, more pressing data on the network. It will be incumbent on the implementation of the protocol to ensure that intervening messages do not disrupt connection mode data and that connection mode data does not disrupt other SAE J1587 message traffic [11]."
- Request To Send Guidance
    - Guidance 4 - "Upon receipt of an RTS, the receiving station must make decisions concerning its ability to buffer the incoming message. If the receiving station cannot accept any

connection mode data it may respond with an ABORT message, signaling that the connection was refused [11]."

- Clear to Send Guidance
  - Guidance 5 - "The receiver may wish to accept the connection request but may not have any resources available to buffer the message at this moment. In this circumstance the receiver shall respond with a CTS indicating the number of segments to be sent to be zero, starting with segment number zero. As segments are numbered from 1 to 255 (FF16), this indicates to the originator that the receiver is amenable to the connection but is at this moment out of resources. When the resources are available, the receiver should transmit a CTS showing the number of segments it can accept, and a beginning segment ID number of 1 [11]."
- End of Message Guidance
  - Guidance 6 – "Used by the receiving station to acknowledge receipt of entire message. Note that this is not strictly needed, if all segments have been acknowledged, the entire segmented message has been received [11]."
- Abort Guidance
  - Guidance 7 - "The connection abort message may be passed by either of the communicating entities if it cannot continue the data transfer process for any reason [11]."

3. Research Questions

Our analysis of these protocol stacks has motivated the present empirical investigation to systematically verify whether vendor implementations satisfy the provided standard guidance. Through a systems engineering lens, we seek to demonstrate how requirement verification through test cases can reveal gaps in current specifications and inform the development of more robust communication systems. Our hypothesis is that a vendor-specific implementation of the standard will not satisfy all the guidance, revealing that additional requirements will be needed to ensure secure and reliable system behavior.

The following research questions guide the structure and development of subsequent sections:

RQ1: To what extent does the Detroit Diesel engine controller implementation satisfy the datalink layer guidance specified in SAE J1708?

This question examines whether the seven guidance statements extracted from the standards are adequately implemented in a production ECU. By developing test cases for bus contention handling (Guidance 1-2), we can verify if the implementation meets the specified behaviors or if gaps exist between the standard's guidance and real-world execution.

RQ2: To what extent does the Detroit Diesel engine controller implementation satisfy the transport protocol guidance specified in SAE J1587?

This question focuses on verifying the connection management (Guidance 3), Request to Send (Guidance 4), Clear to Send (Guidance 5), End of Message (Guidance 6), and Abort (Guidance 7) guidelines through systematic test cases. We examine whether the transport protocol implementation adheres to the specified constraints and behaviors.

RQ3: How can empirical testing of J1708/J1587 implementations inform the development of new systems engineering requirements that address identified gaps in the current standards?

This research question moves beyond vulnerability identification to analyze how our test case results may identify where the current guidance lacks the specificity, or completeness needed for secure

implementations. Additionally, this question focuses on developing new requirements that follow systems engineering best practices for proper requirement engineering principles.

These research questions collectively support our overarching goal of demonstrating how systems engineering requirement principles can be applied to analyze legacy communication protocols and develop requirement-driven improvements for the heavy-truck industry.

4. Verification of Requirements

4.1 Hardware Setup

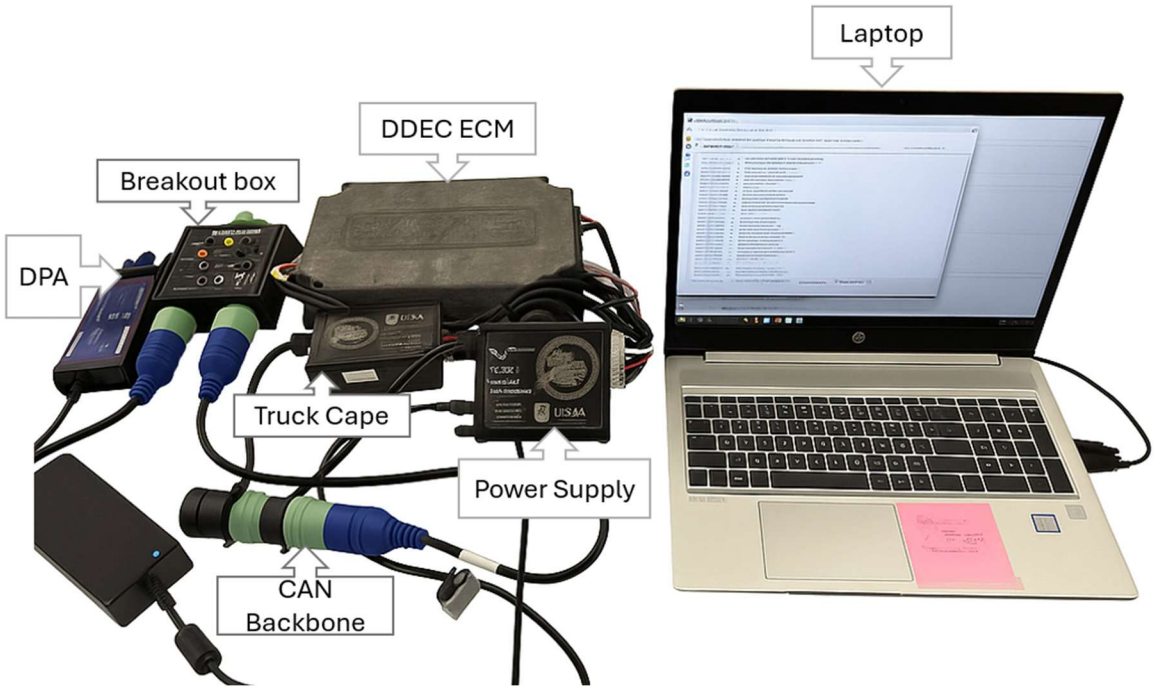

Fig 2. Hardware Setup for J1587/J1708 Testing

The local bench testbed consists of a Series 60 DDEC fourth generation Engine Control Module (ECM) interfaced with supporting hardware, as shown in Fig. 2. The ECM served as the primary unit under test, with diagnostic communication accessed through a RP1210-compliant Dearborn Protocol Adapter (DPA) and Breakout box, which enabled both normal diagnostic sessions and injected traffic. A Truck Cape and CAN backbone provided physical connectivity consistent with in-vehicle wiring. Additionally, the Truck Cape communicates over a J1708 (RS-485, 9600 bps) link and lets us generate accurately timed frames when we run the analysis. To simulate operational conditions without a full vehicle, a Smart Sensor Simulator 2 (SSS2) was included to provide a power supply to the ECM. For visibility and timing accuracy, we capture line-level signals with a logic analyzer and record application-level events/logs on the PC using the RP1210 standard [17]. A laptop running OEM diagnostic software completed the setup, functioning both as a logging station and as the attack platform for message injection and analysis. Together, this configuration allowed controlled experimentation on J1708/J1587 vulnerabilities while preserving the ability to repeat and monitor results under laboratory conditions.

4.2 Testing Methodology

We created tests based on the guidance in section 2.2 to uncover the answers to our first two research questions. Our realistic bench setup enables us to observe both layers of interest to determine how the DDEC ECM behaves during our tests. A preliminary MID-enumeration is used to fingerprint the MIDs that the DDEC ECM responds to, and edge-case tests are derived from the guidance given in the J1708/J1587 standards.

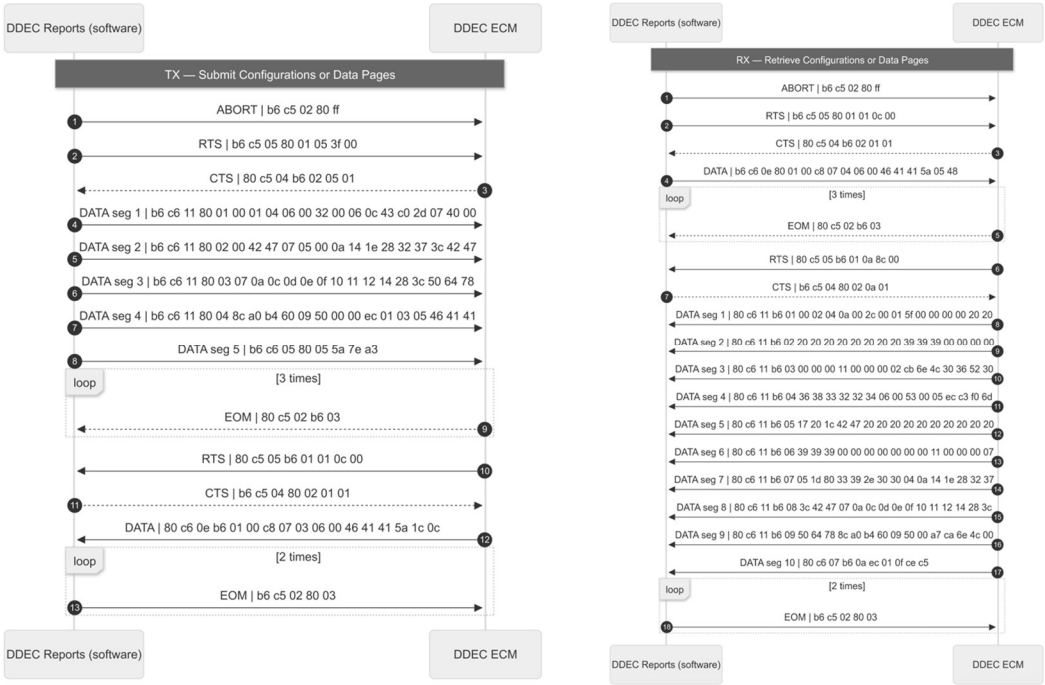

Figure 3. J1587 DDEC Reports Diagnostic Software Normal Transport Protocol Communication: Communication for Extraction of Data Pages (Left) and Submission of Configurations (Right)

Shown in Figure 3, a typical interaction has two parts: the tool sends a small header or setup exchange, and the ECM follows with the data page transfer. After fingerprinting the network, the ECM was found to respond to the following MIDs: 0x80 (Engine #1), 0xB6 (Off-board programming station), and 0xEC (Entry Assist Control #1). The figure also shows two testing interfaces (TX and RX), which allow us to propagate the J1587 Transport Protocol (TP) as either the client (DDEC Reports) or server (DDEC ECM).

Because J1587 is not encrypted, we can parse frames and map out message order, sizes, and inter-message timing. Figure 4 is the communication manager interface used to initiate the transport protocol session between the DPA RP1210 (e.g., DGDPAXL) link and the DDEC ECM that extracts the data page shown in Figure 5.

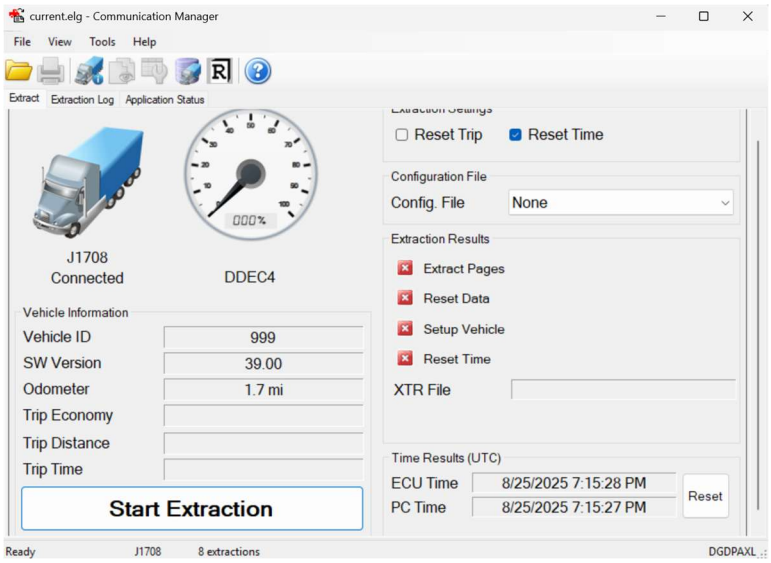

Figure 4. DDEC Reports Communication Manager

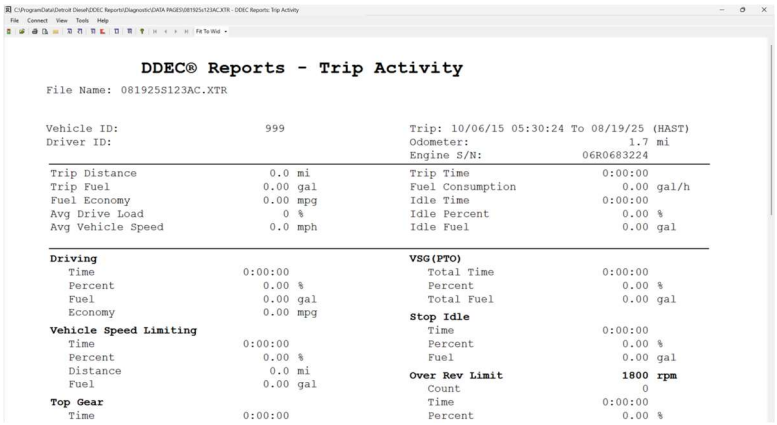

Figure 5. DDEC Reports Data Page

The following sections detail our experiments to verify the guidance imposed by the J1708/J1587 standard. For each test case we justify the relation to the guidance and explain the attack pattern we utilized to determine if the system passed or failed the experiment. We first describe the tests on the datalink and then illustrate the tests on the transport protocol. All testing is on a bench, and the artifacts (raw traces, logs) are kept so others can repeat the setup [16].

4.3 J1708 Spoofing

Guidance 1 serves as a template for the 'J1708 Spoofing' test, developed to verify whether the ECM satisfies J1708's guidance on *permitted* MIDs. Shown through the public repository correlated to this paper, an edge case test was run, spoofing the MID and parameter identification characters (PIDs) of the periodically broadcast message used by the DDEC ECM [16].

- VIN = "CSU" (PID 0xED): "80ED124353550000000000000000000000".
- Make/Model/Serial = "HAS*TAKEN*OVER" (PID 0xF3): "80F312804841532A54414B454E2A4F564552000000"
- Unit Number = "GO" (PID 0xE9): "80E912474F0000000000000000000000000".
- Software ID = "RAMS" (PID 0xEA): "80EA1252414D530000000000000000000000"

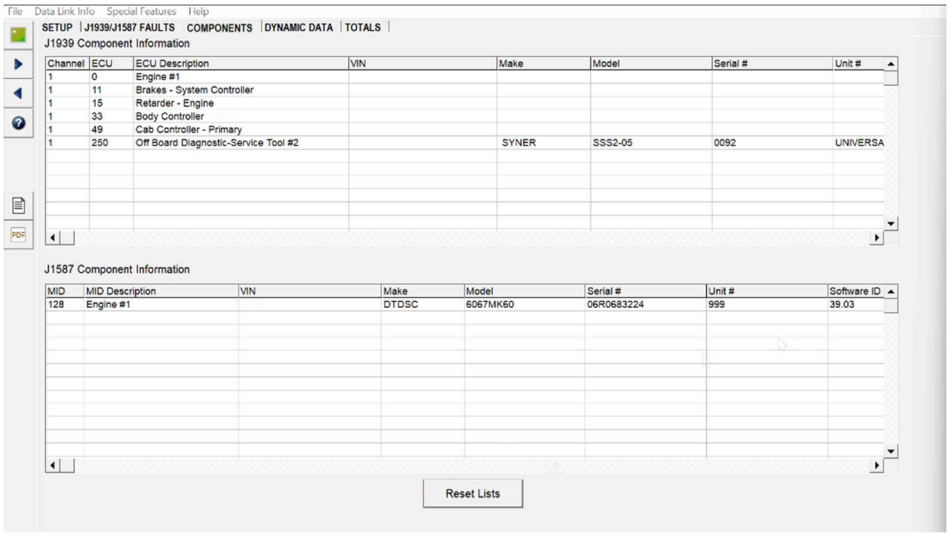

Figure 6. DG Diagnostic RP1210 Link Before Attack

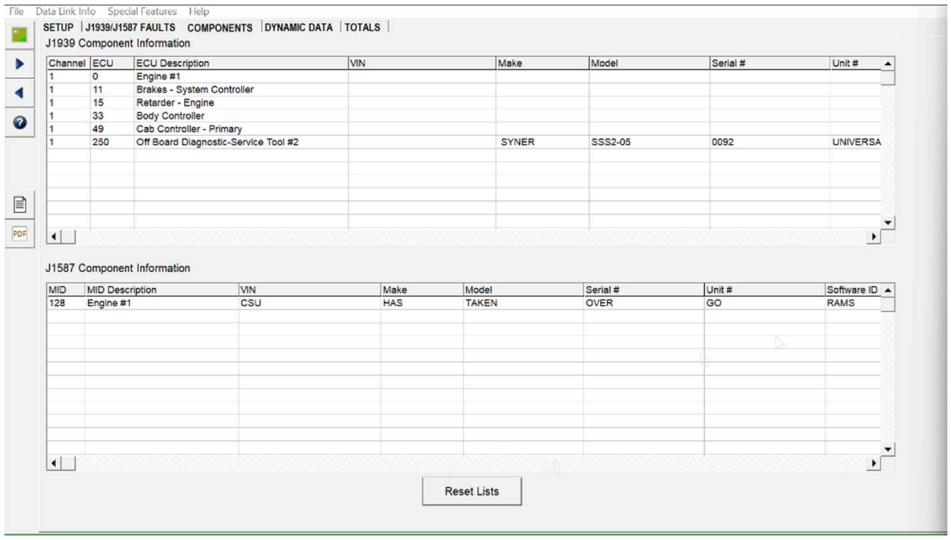

Figure 7. DG Diagnostic RP1210 Link After Attack

Spoofing a MID, or an entire frame, means sending a perfectly valid J1708 message while pretending to be a different ECU. Because the data link layer (i.e., access control layer) lacks authentication or any real source verification, the bus accepts any node that follows the timing and arbitration rules. The result shows up at the application layer, things like bogus diagnostics, falsified sensor readings, or tricking a service tool into believing the fake data (see Figures 6 and 7). Our edge case test highlights a gap in the J1708 guidance: once a device is accepted as a *permitted* MID, that malicious node can still transmit arbitrary messages.

4.4 J1708 "Priority" DoS

The J1708 "Priority" DoS test was developed to verify whether the DDEC system satisfies the Bus Access Time (BAT) from Guidance 2. Bus Access Time (BAT) is a time duration equal to the minimum time of an Idle Line (i.e., 10-bit times or ~1.04 ms) plus the product of 2-bit times and the message priority (which ranges from 1 to 8 [12]. Because J1708 access control is purely timing-based (BAT), an edge case test was crafted given a node that repeatedly transmits short, high-priority messages with inter-message idle less than or equal to the minimum allowed by the standard. Other specific test cases were also crafted to test arbitration of MIDs: one with MID 0x00 and one with MID 0xFF performed against the DDEC Reports diagnostic software.

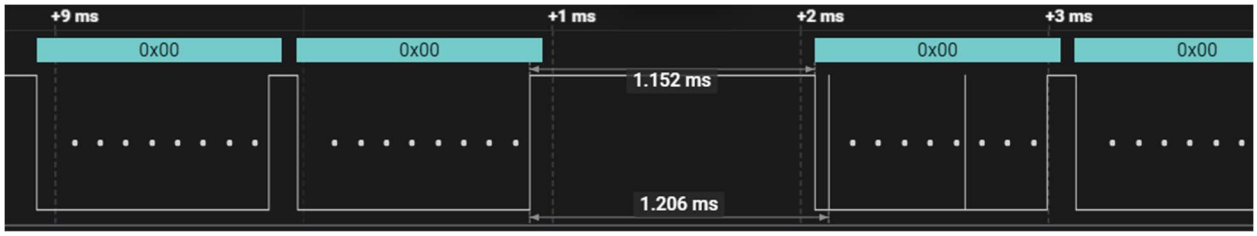

Figure 8. Valid MID 0x00, Valid Checksum 0x00, Surpassing Valid Inter-Frame Gap (BAT) < 1.255 ms

Because J1708 arbitration is bitwise with dominant/recessive signaling, a stream that repeatedly begins sooner (< 1.255 ms at the highest permittable access time), will tend to capture the bus and defer competing traffic. As message length grows, the fraction of time the medium is occupied increases, pushing overall utilization toward saturation. In effect, the timing of UART 8-N-1 provides a pathway to transmit more aggressively than J1708 permits, creating the preconditions for a timing-based denial of service, surpassing MID collisions. Because the ECM satisfies the guidance accordingly, it is immediately kicked off the bus. Our tests highlight a gap in the J1708 Media Access Control (MAC) definitions, specifically priority of critical messages, defined as a *permitted* Bus Access Time [16].

4.5 J1587 Request Overload

The J1587 Request Overload test was developed to verify whether the ECM J1587's guidance satisfies stateful connections determined by Guidance 3 and 4. Because the J1587 Transport Protocol (PID 197) obligates an ECU to parse RTS frames and, when resources appear available, allocate buffers/timers and reply with a CTS or ABORT, a sustained stream of well-formed RTS messages may tie up connection-management state and bus time. We developed a test case to determine if we could overload ECU session resources (buffers, timers, slots, etc.) and crowd out legitimate TP exchanges, leading to delayed or aborted transactions. Using J1708 specified priority for TP traffic (lowest priority), tests were performed by sending repeated RTS messages on the DDEC ECM, shown in Figure 9.

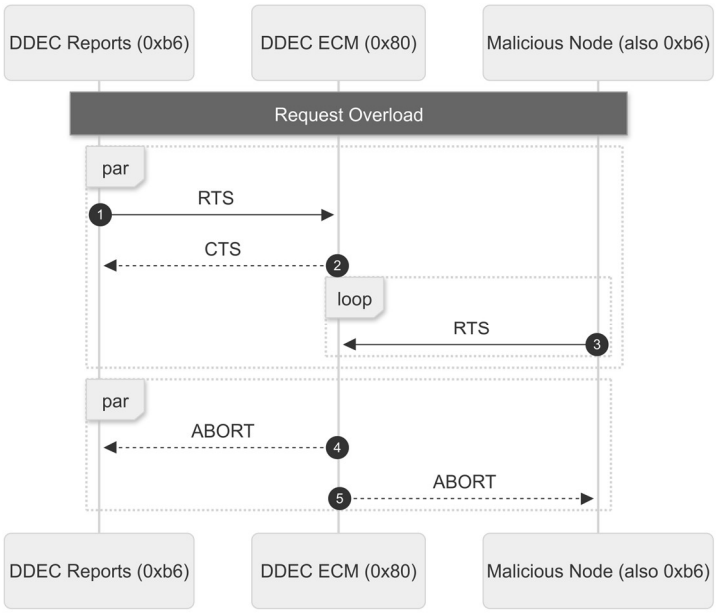

Figure 9. Request Overload Test Results

The test case results show the DDEC ECM successfully passed by sending an ABORT response, signaling that the connection was refused [16]. Although the test passed, it also revealed the requirement for a TP response from the ECU: either an ABORT, or CTS. What was not tested was the speed, or otherwise priority messaging of the TP session (edge-cases), that *could* result in an overload of stateful resources (open sessions, buffers, timers, etc.). Furthermore, the test highlights the importance of the ABORT message as guidance within the J1587 standard. Since either node (only present within that session) can send an ABORT, an attacker posing as a malicious MID (spoofed) *may* be able to effectively force another legitimate connection to halt its data transfer (PID 198).

4.6 J1587 Absent EOM

The J1587 Absent EOM test was developed to verify if the DDEC ECM satisfied the CTS and EOM usage of J1587 Guidance 5 and 6. Based on the J1587 TP (PID 197) guidance, which requires endpoints to allocate buffers/timers and progress connection state upon receiving a CTS, an on-bus adversary that injects well-formed CTS frame can force peers to maintain multiple segmented transfers concurrently. In this way, the Absent EOM test was developed to maintain legitimate connection-management resources and session slots on ECUs/diagnostic tools, delaying or blocking legitimate TP exchanges without using malformed frames. Using J1708 specified priority for TP traffic (lowest priority), tests were performed by sending repeated CTS messages on the DDEC ECM, shown in Figure 10.

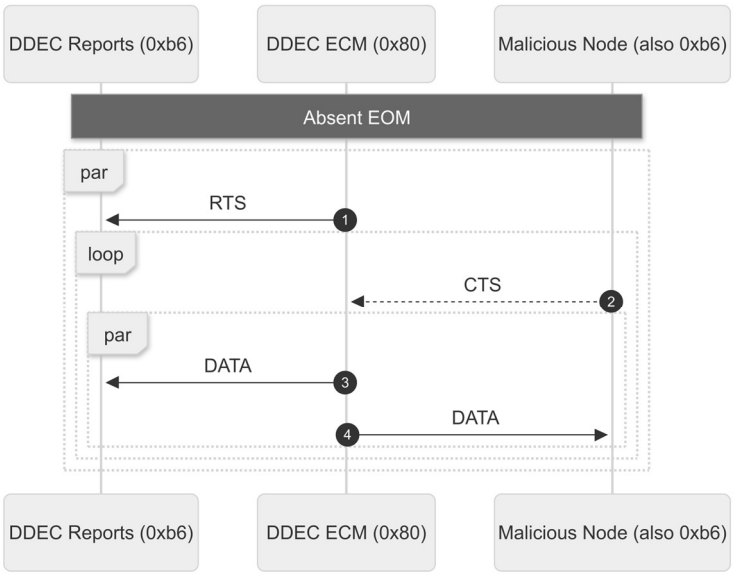

Figure 10. Absent EOM Test Results

The test results show the DDEC ECM enabled the reusage of CTS messages as detailed in the standard but without an EOM present, the state of the targeted node is in a concurrent TP state, overflowing the TP session with DATA sent after a targeted CTS reinitializes the session. Furthermore, edge-cases for nodes outside of the current session, also in a legitimate TP session, were found to be unresponsive, indicative of a connection-exhaustion attack [16].

4.7 Abort Misuse

The J1587 Abort test was developed to verify Guidance 7 and to evaluate whether the ABORT command could be misused to prematurely terminate an active transport session on the Detroit Diesel ECM. While it is intended as a recovery mechanism when an ECU cannot complete a transfer, it also provides an adversary with an opportunity to deliberately disrupt a connection. Since J1587 lacks authentication or integrity checks, any node that can convincingly spoof a valid participant could potentially inject ABORT messages. The edge-case test involves repeatedly injecting ABORT messages to force termination of any node within a TP session. Figure 11 illustrates this interaction where the session begins with RTS and CTS progressing into data transfer and is then prematurely terminated by an injected ABORT.

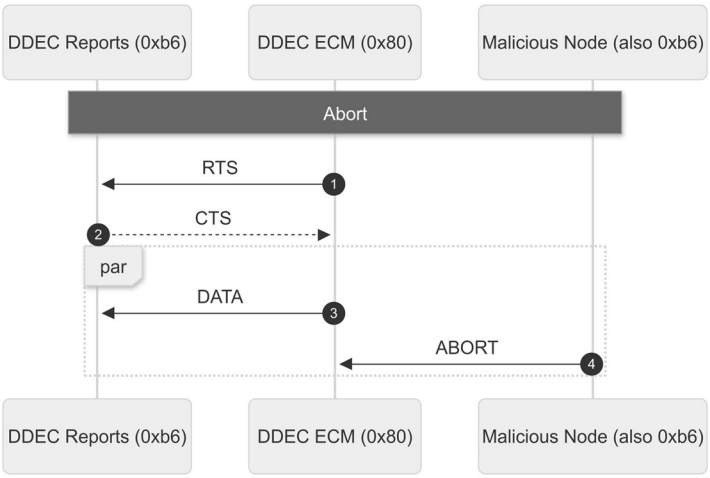

Figure 11. Abort Test Results

The results show that the cumulative occurrences over time demonstrate how injected ABORT messages from the spoofed off-board tool MID (0xB6) consistently terminate sessions, forcing the ECM to repeatedly reset. This behavior confirms that the ECM accepts ABORT from a valid peer and compiles with Guidance 7. However, it also shows that an attacker imitating a valid peer can easily exploit this mechanism to deny service. Additionally, when ABORT was injected by an unrelated ECU MID 0xEC the DDEC ignored the request which highlights that the implementation enforces a peer-to-peer model. This leaves the possibility of a persistent DoS attack against diagnostic tools or ECUs highlighting the need for stronger requirements such as validating the origin of ABORT messages.

5. Discussion

Quality requirements are foundational in the development of complex systems and strong systems engineering utilizes requirements engineering throughout the product lifecycle [15]. However, the requirement development and requirement management of the complete standards are outside the scope of this research. We are only concerned with the aspects of requirements engineering for analyzing, verifying, and developing requirements for a handful of susceptible areas in the datalink layer and transport protocol. As a result, we did not transform the SAE standards terminology into proper systems engineering requirements, but we analyzed and verified them with test cases.

RQ1 stated as to how well the DDEC ECM adheres to SAE J1708's datalink layer standards. The ECM did, in fact, adhere to the standards, as demonstrated by the outcomes of the spoofing and denial-of-service tests. As required by the standard, arbitration rules were followed, and each message started with a valid MID. The system met the J1708 datalink standards in terms of compliance. These studies did, however, also demonstrate that the existing guidance is not robust to prevent misuse. The DoS test revealed that a node transmitting faster than the permitted bus access time could control the channel and prevent legitimate traffic, while the spoof test showed that any node presenting a valid MID could inject arbitrary data without being challenged. As a result, even if RQ1 has an affirmative response in terms of compliance, the result also shows that security is not the same as compliance.

RQ2 states as to how well the DDEC ECM adheres to SAE J1587's transport protocol guidelines. The abort, request overload, and absence EOM trials verified that the ECM acted as expected by the standard. It processed Abort messages in accordance with the guidelines, managed RTS/CTS windows, and enforced

single-connection restrictions. Once more, the ECM met the requirements of the transport protocol and even additionally prevented multiple RTS messages in an open session. However, the findings exposed exploitable flaws that the standard fails to address. Since explicit closure was not necessary in the event of the absence EOM, sessions could continue forever. Additionally, an attacker can end sessions whenever because inserted Abort messages were accepted without validation in the abort test.

When combined, the responses to RQ1 and RQ2 demonstrate that the DDEC ECM met the requirements of SAE J1708 and J1587 recommendations; nonetheless, exploitable behaviors were still found in each instance. This indicates no evidence is available to support the initial theory vendor implementations would not meet the criteria. Rather, the results show that more requirements are required to fix the gaps in the standards themselves. RQ3 concentrates on establishing requirement-driven mitigations directly impacted by this study's results.

5.1. Proposed Requirements

This section directly answers RQ3 because it illustrates how the gaps discovered during verification testing are transformed into requirements for more secure and complete implementations. This study emphasizes the significance of defining proactive requirements in mitigating message management type imparities in J1708 and J1587. Following systems engineering principles we connect the gaps to system needs and then transform them into specific structured requirements [15].

INCOSE Guide to Writing Requirements (GtWR) offers characteristics and rules for well-formed sets of needs and requirements. It defines a requirement statement as "the result of a formal transformation of one or more sources, needs, or higher-level requirements into an agreed-to obligation for an entity to perform some function or possess some quality within specified constraints with acceptable risk" [8]. This study has compiled the relevant sources and proven the necessary data to detail the need for requirements that specify the protection against the J1708/1587 protocol gaps.

The need statements below are connected to the results detailed in the previous sections.

- The ability for ECUs to check for message integrity. (Results from the J1708 Spoof Test and J1587 Abort Test)
    - *Note: Many issues arise because this need statement isn't met.
- The ability for ECUs to detect and isolate nodes exhibiting abnormal transmission behavior. (Results from the J1708 "Priority" DoS test)
- The ability for ECUs to ignore request messages. (Results from the J1587 Request Overload Test)
- The ability for ECUs to close open connections after a certain amount of time if the connected ECUs are no longer exchanging data. (Results from the J1587 Absent EOM Test)

The requirements we created will all have elements of the [WHO] shall (WHAT) <HOW WELL> under {CONDITION} to conform with a uniform structure that improves their quality [15]. Additionally, we followed INCOSE's Formal Transformation and the Agreed-to Obligation rules and characteristics to develop all the requirements. Highlighting the characteristics C1 Necessary, C3 Unambiguous, C5 Singular, C7 Verifiable, and C9 Conforming [8].

1. The [system] shall (validate the integrity) of a received message by <verifying its accompanying secret verification code, resulting in the rejection of 100% of messages that fail validation> {before processing its contents.}

2. The [system] shall (ignore all messages) <from any single MID that transmits more than (n) high-priority messages> {within a (t)-second time window.}
3. The [system] shall (restrict the number) of {active RTS requests per node} <to a maximum of (x)>.
4. The [system] shall (limit the maximum duration of any RTS/CTS modes) <to (d) seconds>, {regardless of activity.}

These requirements now account for the tested disparities in the standard. Ultimately this process can aid systems engineers and the heavy-truck industry in their discussions for secure by design systems.

6. Conclusion

Legacy diagnostic protocols such as SAE J1708 and J1587 remain widely deployed in mixed fleets despite lacking authentication, and session lifecycle controls proposed in this paper. Prior research has demonstrated transport-layer exploits in J1939 and proposed prototype intrusion detection for J1708/J1587, but the vulnerability of J1587 itself under adversarial conditions has not been systematically validated. This paper presents an empirical study of J1708/J1587, focusing on its data link and transport/session layers. This study represents the first systematic validation of J1708/J1587 vulnerabilities and their translation into actionable requirements. These results show that the standard would benefit from more requirements that handle misuse of the protocols.

We propose a set of requirement-driven protections for the integrity of messages, identifying abnormal behavior of RTS and high priority messages, and transport protocol session time enforcements. These requirements provide a vendor-neutral framework that allows systems engineers to constrain exploitable behaviors and ensure more predictable and trustworthy communication management systems. By grounding protocol analysis in experimental evidence and framing security enhancements at the requirement level, this work contributes a path forward for securing current and future protocols and offers a foundation for future research into standardization and stakeholder adoption.

6.1 Future Work

This study uses a systems engineering approach to analyze the J1708/J1587 standard, presenting the results as a roadmap for enhancing future heavy-truck communication systems. However, a primary limitation of this study is the restricted OEM coverage. While the intent was to validate vulnerabilities across a wide range of ECUs, not all diagnostic software was available for use. As a result, findings confirm the gaps but cannot claim complete representation of every vendor implementation.

Moving forward, our research will expand in three key directions. First, we plan to extend testing to a broader range of OEM platforms. This will require acquiring and configuring additional proprietary diagnostic software so that attack scenarios can be replicated across multiple vendor implementations. A more diverse set of ECUs will provide greater insight into how different manufacturers satisfy the guidance requirements and will help establish patterns of resilience and weakness across the industry.

Second, we intend to transition from bench-level validation to real truck testing. Running experiments in a full vehicle environment will allow us to capture the complexity of cross-ECU interactions, network load dynamics, and operational stresses that cannot be reproduced on isolated modules. This step is essential for understanding the practical impact of J1587/1708 vulnerabilities on diagnostics and vehicle availability in real-world conditions.

Thirdly, the authors propose a case for digitizing standards into Model Based Systems Engineering tools for easier analysis of requirements written in the documents. After combing through the standards for

ambiguous verbiage the authors conclude that digitization of the requirements would enable the creation of test cases for improved and more secure implementations.

Together, these directions will advance this work from controlled demonstrations toward a more comprehensive and field-validated framework for securing legacy diagnostic protocols.

References

[1] Bernin, F., Butler, M., Cansell, D., Hallerstede, S., Kronlöf, K., Krupp, A., ... & Cansell, D. (2004). Formal Modelling of Electronic Circuits Using Event-B: Case Study: SAE J1708 Serial Communication Link. UML-B Specification for Proven Embedded Systems Design, 211-226.

[2] Saastamoinen, R. (2008). Analysis of the SAE J1708 protocol. Mälardalen University. Västerås, Sweden.

[3] Kumar, S., Daily, J., Ahmed, Q., & Arora, A. (2023). Cybersecurity Vulnerabilities for Off-Board Commercial Vehicle Diagnostics. SAE International Journal of Advances and Current Practices in Mobility, 5(2023-01-0040), 2393-2404.

[4] Nnaji, D., & Daily, J. (2024). *Trucking Forward: Intrusion Detection for SAE J1708/J1587 Networks in Heavy-Duty Vehicles* (No. 2024-01-2805). SAE Technical Paper.

[5] Boland, H. M., Burgett, M. I., Etienne, A. J., & Stwalley III, R. M. (2021). An overview of can-bus development, utilization, and future potential in serial network messaging for off-road mobile equipment. Technology in Agriculture.

[6] Society of Automotive Engineers, "SAE J1939 Standards Collection." [Online]. Available: https://www.sae.org/standardsdev/groundvehicle/ j1939a.htm

[7] Chatterjee, R., Mukherjee, S., & Daily, J. (2023). Exploiting transport protocol vulnerabilities in SAE J1939 networks. In *Proceedings of the Inaugural International Symposium on Vehicle Security & Privacy*. Internet Society.

[8] Ryan, M., & Wheatcraft, L. (Authors). (2023). INCOSE Guide to Writing Requirements V4 – Summary Sheet. International Council on Systems Engineering (INCOSE).

[9] Murvay, P. S., & Groza, B. (2018). Security shortcomings and countermeasures for the SAE J1939 commercial vehicle bus protocol. IEEE Transactions on Vehicular Technology, 67(5), 4325-4339.

[10] Lee, Hwejae, et al. "Expanding the attack scenarios of sae j1939: A comprehensive analysis of established and novel vulnerabilities in transport protocol." arXiv preprint arXiv:2406.00810 (2024).

[11] "Electronic Data Interchange Between Microcomputer Systems in Heavy-Duty Vehicle Applications(STABILIZED Jan 2013)" available online at https://www.sae.org/standards/content/j1587_201301/

[12] Serial Data Communications Between Microcomputer Systems in Heavy-Duty Vehicle Applications(STABILIZED Sep 2016) J1708_201609 available online at https://www.sae.org/standards/content/j1708_201609/

[13] Mukherjee, S., Shirazi, H., Ray, I., Daily, J., & Gamble, R. (2016, November). Practical DoS attacks on embedded networks in commercial vehicles. In International Conference on Information Systems Security (pp. 23-42). Cham: Springer International Publishing.

[14] International Organization for Standardization. (1994). Information technology — Open systems interconnection — Basic reference model: The basic model (ISO/IEC 7498-1:1994). https://www.iso.org/standard/20269.html

[15] Wheaton, J. S., & Herber, D. R. (2024). Digital requirements engineering with an INCOSE-derived SysML meta-model. In D. Verma, A. Gorod, & C. J. J. Paredis (Eds.), The Proceedings of the 2024 Conference on Systems Engineering Research. Conference on Systems Engineering Research Series. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-62554-1_2

[16] SystemsCyber. (2025, August). hv_legacy_vulnerabilities. GitHub. Retrieved August 1, 2025, from https://github.com/SystemsCyber/hv_legacy_vulnerabilities

[17] Kvaser. (n.d.). Introduction to RP1210A (RP1210B). Kvaser. Retrieved September 1, 2025, from https://kvaser.com/about-can/can-standards/rp1210/