On In-Vehicle Network Security Testing Methodologies in Construction Machinery

Sheela Hariharan*,
Mälardalen University, Västerås, Sweden
Volvo Construction Equipment, Eskilstuna, Sweden
sheela.hariharan@mdu.se

Alessandro V. Papadopoulos
Mälardalen University
Västerås, Sweden
alessandro.papadopoulos@mdu.se

Thomas Nolte
Mälardalen University
Västerås, Sweden
thomas.nolte@mdu.se

***Abstract*—In construction machinery, connectivity delivers higher advantages in terms of higher productivity, lower costs, and most importantly safer work environment. As the machinery grows more dependent on internet-connected technologies, data security and product cybersecurity become more critical than ever. These machines have more cyber risks compared to other automotive segments since there are more complexities in software, larger after-market options, use more standardized SAE J1939 protocol, and connectivity through long-distance wireless communication channels (LTE interfaces for fleet management systems). Construction machinery also operates throughout the day, which means connected and monitored endlessly. Till today, construction machinery manufacturers are investigating the product cybersecurity challenges in threat monitoring, security testing, and establishing security governance and policies. There are limited security testing methodologies on SAE J1939 CAN protocols. There are several testing frameworks proposed for fuzz testing CAN networks according to [1]. This paper proposes security testing methods (Fuzzing, Pen testing) for in-vehicle communication protocols in construction machinery.**

***Index Terms*—construction machinery, SAE J1939, Fuzzing, pen testing, security testing**

I. Introduction

The complexity of in-vehicle communication networks increases as the number of ECUs and external connectivity increases due to user demands. In fact, there are examples of vehicles with over 100 ECUs connected with several complementing communication technologies [2]. Although there is a recent change in internal communication with high data rates using Automotive Ethernet and advancements in zonal architectures [3], still most used and established communication is the Controller Area Network (CAN) [4].

SAE J1939 is a higher layer protocol that is based on CAN. Over the past decades, SAE J1939 has evolved into and has been accepted as the industry standard for heavy-duty vehicles such as trucks and construction machines. The main issue in the SAE J1939 protocol is, similar to the CAN standard, that security has not been considered during the design of the protocol [5]. In fact, the messages can be decoded easily on the network by intruders since the message format and the IDs that are used, as per standard, only requires access to the standard documents. Hence, typically, no significant effort of reverse engineering is required.

The work presented in this paper is supported by the Swedish Knowledge Foundation (KKS) via the research project ARRAY++.

Today, as systems implement a higher degree of external connectivity, the security of such systems is instrumental [6]. Murvay et al. [7] presented a list of recently reported attacks on J1939, highlighting some of the main security shortcomings of the protocol. This paper provides an overview of security testing approaches relevant to construction machinery, including pen testing, fuzzing, and several other testing methods that can identify all possible vulnerabilities in the machine ecosystem. The aim of this work-in-progress paper is as follows

1) to provide an overview of automotive security testing,
2) to propose a testing method that covers pen testing and fuzzing, and
3) to implement a testbed that covers integrating the testing in a HIL/vehicle environment.

A. Outline

The outline of this paper is as follows. In Section II we present the background of our work, including an overview of the SAE J1939 standard followed by an overview of central concepts of security testing. In Section III we go into the details of security testing within the application domain of in-vehicle networks of construction machines. Section IV presents our ongoing work-in-progress regarding testing utilizing fuzz testing. Finally, Section V concludes the paper.

II. Background

A. The SAE J1939 protocol

SAE J1939 provides serial data communications between Electronic Control Units (ECU) in any kind of commercial or heavy-duty vehicle. The SAE J1939 message format is shown in Figure 1. The messages exchanged between these units can be data such as the vehicle road speed, the torque control message from the transmission to the engine control unit, and other relevant signals.

SAE J1939 is a primarily data-driven protocol. In fact, SAE J1939 provides a far better data bandwidth than any of these automation protocols, and it is commonly used in heavy vehicles [8], [9]. J1939 data packets contain the actual data and a header, which contains an index called Parameter Group Number (PGN). A PGN identifies a message's function and associated data. J1939 attempts to define standard PGNs to encompass a wide range of automotive, agricultural, marine and off-road vehicle purposes [10]. SAE J1939 and classical

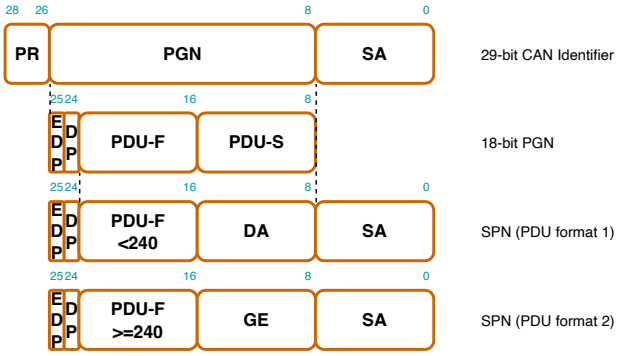

Fig. 1. The SAE J1939 Message Format.

CAN (along with CANFD) differ mainly with Higher Later Protocols(HLPs) and the type of applications where it is used according to [11]

B. Overview of Security Testing

The vehicle safety standard ISO 26262 supports the automotive lifecycle of construction machines, and the SAE/ISO 21434 standard defines related cyber-security standards. The interplay between these standards is becoming more important. Since finding a vulnerability affects the safety of the vehicle, security testing is essential to be integrated into the Application Lifecycle Management (ALM). It also includes incident reporting and having a strategy for incident handling. Most of the testing is performed manually like acceptance testing of the product. Security-related testing should have limited manual intervention and should be more automated. Fuzzing is one of the dynamic methods which can be automated efficiently using the existing resources like Simulation-In-the-Loop (SIL) and Hardware-In-the-Loop (HIL) setups [12].

1) Vulnerability Scanning: When the System Under Test (SUT) is partially developed and able to provide some functionality, vulnerability scanning can be performed. It is a dynamic scalable approach to send various inputs to the software functions of the system and examine the patterns generated as the outputs [13]. There are several vulnerability scanning tools available to send such inputs to the system for the known threat vectors. These known attacks can be derived from TARA (Threat And Risk Assessment) and then the vulnerability scanning tool is configured for such attacks to observe how the target system behaves [14]. The tool also focuses on finding the weakness for unknown vulnerabilities also, if we know the known attack patterns. This scanning is helpful in finding the configuration mishandling and miss some input arguments in the software functions.

2) Penetration testing: Security engineers act as ethical hackers for validating the resilience of vehicle software systems against malicious behaviors and attempts to penetrate the system under test. Penetration testing is the most researched and widely used testing methodology in the automotive industry [1]. There are many researchers and experiments conducted on several kinds of physical and remote attacks, also including testing frameworks that can bypass the intrusion detection systems installed in the vehicle [15]. While penetration testing generates the most signihelps find manual-driven and the most time-consuming, and often requires deep domain expertise. Automating known attacks is always an essential factor of a functional penetration testing strategy. Good coverage of well-known issues and attacks, as well as the most likely and significant attacks, can be reasonably well covered through penetration testing. However, it is not enough to conduct penetration testing to ensure the resilience of vehicle software systems.

3) Fuzzing: Fuzz testing is the most scalable approach for testing that checks for unexpected behaviors in the SUT by giving random inputs. Fuzzing is an automated test procedure used in cyber-security to mimic a potential 'cyber-attack' and determine those vulnerabilities before the product launch. The fuzz testing can be carried out as white box, gray box or black box fuzzing [1]. Testers mainly focus on black-box fuzzing due to the complexity of automotive software as performing white-box will require more effort and time consuming along with costs. Since the automotive industry has a vast supply chain having software developed by tier suppliers performing white box testing by the automotive manufacturer will not be practical.

III. Security Testing for Construction Machines In-vehicle Network

A. UDS Fuzzing and Pen testing on SAE J1939 CAN bus

Over the years there have been improvements in the cyber-security standards that are used in the automotive industry [16], such as SAE J3061, ISO 21434, and UNECE R155/R156. However, all these standards and regulations only give recommendations and they provide high-level guidance for how cyber-security testing should be performed. This generalization ends up with having different implementations and testing methodologies.

The main objective of cyber-security testing is to identify the security vulnerabilities and to, in the end, have secure functions. On the other hand, there are only a few specific automotive-related security testing tools. There are many open-source security tools available nowadays like Burpsuite, Nmap, etc. but they do not cover the entire portfolio of automotive network protocols. Those tools can be used only for a few systems, e.g., the infotainment system, which have Ethernet and WiFi connectivity, and they are not suited for other widely used protocols such as SAE J1939 CAN, LIN, and Flexray. It is equally important to test and protect such in-vehicle networks, as cyber-attacks on those result in affecting the functionalities of safety-critical systems.

IV. Work-in-progress

In this section, we present our current work-in-progress towards a framework for cyber-security testing suitable for software systems inherent in the construction industry.

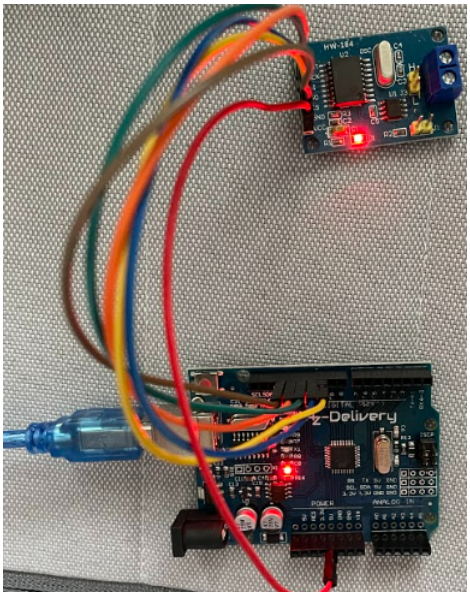

Fig. 2. Test setup using an Arduino controller that acts as an intruder ECU (the hacker) on the CAN bus.

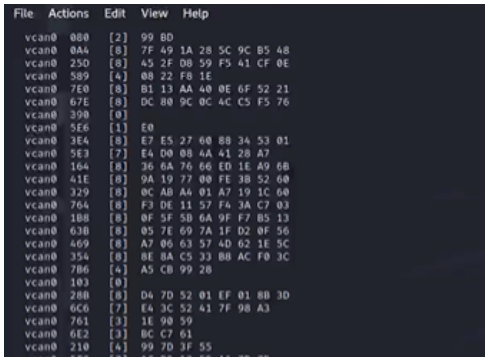

Fig. 3. Message reception configuration.

A. Test setup

The test setup (the testbed) shown in Figure 2 is as follows:

1) one simulated ECU running on a virtual machine platform using Linux, and
2) one intruder ECU (the hacker) that is using an Arduino microcontroller together with an Arduino MCP 2515 CAN transceiver.

This testbed experiment is to carry out the penetration testing for the SAE J1939 higher-layer CAN that is used in construction machines' in-vehicle networks, along with UDS communication that is used for diagnostics. CAN open source tools on Kali Linux are used to send and receive the messages on the CAN transceiver [17]. The first step is to identify the UDS-supported ECUs on the network. Later, the ping/request is sent on the network from the Linux environment (the simulated ECU) or from the test ECU that is being developed. Then the screening process is done from either Linux (the simulated ECU) or from the Intruder ECU, to search for any open vulnerabilities. This is done by analyzing the responses

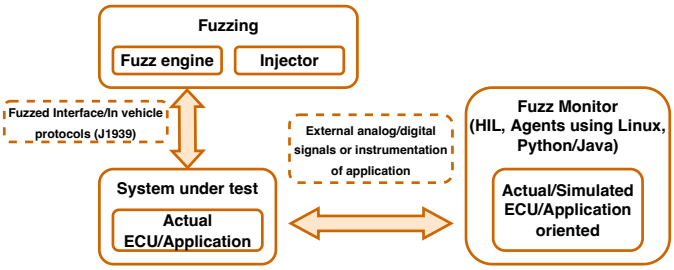

Fig. 4. Example of a fuzz testing environment.

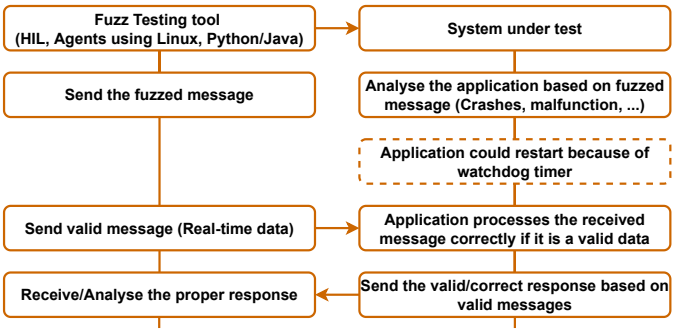

Fig. 5. Example of fault detection with fuzzing.

to the UDS requests sent in the network. Some requests need security access, and breaking the security by brute force is one method that is also a possible option within the setup of this experiment.

Fuzzing can be done on UDS services using this setup, where numerous requests are sent from the fuzzer, System under test(SUT) environment example in Figure 4 to search for any UDS implementation vulnerabilities in the diagnostics manager function. The tester functions can perform various UDS requests for different services, mutating the data and then sending the random data on the CAN bus. During the process, the sniffer is used to monitor the data processing for any abnormal data on the bus.

B. Preliminary results

Looking at initial results that have been extracted from the UDS fuzzing, sending a sequence of requests on the bus according to Figure 5 testing sequence, includes a no response from one of the ECUs on the CAN bus, which shows that the ECU has crashed and a scanning vulnerability has been found. The system went back to normal when a reset is performed.

This testing framework approach is planned to extend for fuzzing other connectivity protocols like USB, Bluetooth, and WiFi, and examine the corresponding behavior. This could be a potential solution for finding vulnerabilities inherent in remote attacks. The future work also covers approaches to automate the security testing methods together with Machine learning and other possibilities.

V. Conclusions

The work needed to find all hidden vulnerabilities in terms of connectivity as well as vulnerability inherent in the communication protocols of the CAN bus is increasingly more tedious since the number of methods to intrude into a network is increased by hackers every day. At the end of the day we cannot control all unknown vulnerabilities, however, the best way to find and prevent them is by having proper security testing integrated into the product development lifecycle. Having in place automated testing using, e.g., fuzzing, is potentially an efficient solution when it comes to finding new threats and which allows us to take measures by, e.g., providing solution patches later.

This paper proposes a testing framework, developed for construction machines' in-vehicle networks, where the same framework potentially can be utilized by most systems within the heavy vehicle industry. This test bed is currently a work-in-progress setup for finding unknown vulnerabilities which are not covered by other security testing mechanisms. Moreover, the framework is performed as a black box, and it utilizes methods to integrate with having a framework compatible with HIL systems. The main advantage of this working setup is to not only be limited to finding the vulnerabilities but also analyzing the results and proposing mitigation methods for such threats. This is potentially very helpful for the software developers and it can be used for fixing software. Later it can be used to build software Over The Air (OTA) updates.

Acknowledgments

The work presented in this paper is supported by Volvo Construction Equipment AB, Mälardalen University, and the Swedish Knowledge Foundation (KKS) via the research project ARRAY++.

References

[1] L. J. Moukahal, M. Zulkernine, and M. Soukup, "Vulnerability-oriented fuzz testing for connected autonomous vehicle systems," *IEEE Transactions on Reliability*, vol. 70, no. 4, pp. 1422–1437, 2021.

[2] T. Nolte, H. Hansson, and L. L. Bello, "Automotive communications - past, current and future," in *Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA05)*. IEEE Industrial Electronics Society, September 2005, pp. 985–992. [Online]. Available: http://www.es.mdh.se/publications/766-

[3] D. Stöhrmann, A. Kostrzewa, R. Ernst, and H. Kellermann, "Towards ofdma-based ethernet for future in-vehicle communication," in *2020 2nd International Conference on Societal Automation (SA)*, 2021, pp. 1–8.

[4] ISO 11898-1, "Road Vehicles - interchange of digital information - controller area network (CAN) for high-speed communication," in *ISO Standard-11898, International Standards Organisation (ISO)*, Nov. 1993.

[5] C. Jichici, B. Groza, R. Ragobete, P.-S. Murvay, and T. Andreica, "Effective intrusion detection and prevention for the commercial vehicle SAE J1939 CAN bus," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2022.

[6] A. Čaušević, A. V. Papadopoulos, and M. Sirjani, "Towards a framework for safe and secure adaptive collaborative systems," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, Jul. 2019, pp. 165–170.

[7] P.-S. Murvay and B. Groza, "Security shortcomings and countermeasures for the SAE J1939 commercial vehicle bus protocol," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4325–4339, 2018.

[8] Y. Burakova, B. Hass, L. Millar, and A. Weimerskirch, "Truck hacking: An experimental analysis of the SAE j1939 standard," in *10th USENIX Workshop on Offensive Technologies (WOOT 16)*. Austin, TX: USENIX Association, Aug. 2016. [Online]. Available: https://www.usenix.org/conference/woot16/workshop-program/presentation/burakova

[9] C. Jichici, B. Groza, R. Ragobete, P.-S. Murvay, and T. Andreica, "Effective intrusion detection and prevention for the commercial vehicle SAE J1939 CAN bus," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2022.

[10] W. Voss, *A Comprehensible Guide to J1939*. Copper Hill Media Corporation, 2008.

[11] Sae j1939 vs. can bus - what's the difference? [Online]. Available: https://copperhilltech.com/blog/sae-j1939-vs-can-bus-whats-the-difference/

[12] D. Oka. Defensics fuzz testing. [Online]. Available: https://www.synopsys.com/software-integrity/security-testing/fuzz-testing.html

[13] E. F. M. Josephlal and S. Adepu, "Vulnerability analysis of an automotive infotainment system's WIFI capability," in *2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)*, 2019, pp. 241–246.

[14] Putting automotive security to the test - escrypt. [Online]. Available: https://www.escrypt.com/en/news-events/automotive-security-testing

[15] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 447–462.

[16] SAE International, *J3061: Cybersecurity guidebook for cyber-physical vehicle systems*. http://standards.sae.org/j3061 201601/, 2016.

[17] C. Smith, *The car hacker's handbook: a guide for the penetration tester*. No Starch Press, 2016.