Article citation info:
Pełechaty P, Konieczny Ł. Analysis of security vulnerabilities in vehicle on-board diagnostic systems. Diagnostyka. 2024;25(3):2024310. https://doi.org/10.29354/diag/192162.

DIAGNOSTYKA, 2024, Vol. 25, No. 3

e-ISSN 2449-5220
DOI: 10.29354/diag/192162

ANALYSIS OF SECURITY VULNERABILITIES IN VEHICLE ON-BOARD DIAGNOSTIC SYSTEMS

Piotr PEŁECHATY *, Łukasz KONIECZNY

Silesian University of Technology, Faculty of Transport and Aviation Engineering,
Krasińskiego 8 str., 40-019 Katowice, Poland
* Corresponding author, e-mail: piotr.pelechaty@polsl.pl

Abstract

The article explains the different types of on-board diagnostic systems (OBD) used in motor vehicles, as well as the impact of the latest automotive security norms on diagnostic interface security. The paper focuses on identifying potential security threats in on-board diagnostic systems used in automotive control units. During the research, a diagnostic interface device of its own design, carrying out special test procedures, was excavated. The research was conducted on several vehicles and ECUs, applying black box penetration testing. The paper's goal is to list all identified vulnerabilities in diagnostic protocol implementation and suggest some corrective actions for software that would increase security. The authors defined a list of low-cost software requirements that can be easily implemented on most modern ECUs.

Keywords: OBD, diagnostic, UDS, cybersecurity, penetration testing

List of Symbols/Acronyms

CAN – Controller Are Network;
DLC – Data Link Connector;
ECU – Electronic Control Unit;
NVM – Non-Volatile Memory;
OBD – On-Board Diagnostic;
SA – Security Access;
UDS – Unified Diagnostic Services;
WWH-OBD – World Wide Harmonized OBD.

1. INTRODUCTION

The use of sophisticated on-board diagnostic systems (OBD) integrated into electronic control units (ECUs) is required to preserve the reliability of complex mechatronic systems, such as those found in currently manufactured motor vehicles. In order to assist with creating a diagnostic hypothesis regarding the condition of electronic systems and to identify malfunctions, a vehicle's OBD systems facilitate the exchange of data with external tester devices [1]. Under the common name "OBD II," emissions-related systems were standardized to lessen their negative impact on the environment. In addition to the mandatory OBD functions, all vehicle interiors were equipped with a standard Data Link Connector (DLC) that enabled the tester device's physical connection [2]. Moreover, under the UDS (Unified Diagnostic Services) protocol, automakers are adding their own extended diagnostic functionalities on top of the mandated and standard OBD systems.

Every tester can access the standard OBD services, which are clearly defined by the norm [1]. Standard OBD services are utilized not only by repair shops but also, for instance, by automobile rental companies and is often used for research purposes [3, 4]. The ability to understand the meaning and implications of the extended UDS services, on the other hand, is exclusive to automakers. The extended UDS diagnostic services are used for advanced procedures like updating the ECU's software, calibration, initializing during the production process, reading confidential information, and performing tests on actuator components.

Both the extended UDS and standard OBD diagnostics use the same communication channel, but they have different applications, target groups, and security requirements. To access every ECU within the car, OBD and UDS frequently utilize the same DLC connector and CAN bus. According to the road vehicle diagnostic norm, the enhanced UDS diagnostics can make use of the standard CAN identifiers designated for legally mandated OBD as long as they don't interfere with others [5]. Furthermore, the OBD protocol is becoming outdated and will be replaced by more recent standards like WWH-OBD or OBDonUDS. This means that the UDS protocol will be used to implement both the standard OBD services and these

Received 2024-06-12; Accepted 2024-08-08; Available online 2024-08-09

expanded ones, and the security of separation will be even more necessary.

1.1. Cybersecurity services of on-board diagnostic systems

Ensuring a clear separation between enhanced UDS services and public OBD services is essential for cybersecurity reasons. Automotive cybersecurity has become increasingly important in recent years as vulnerabilities in automotive systems have come to light. Onboard diagnostics in cars are an important aspect of cybersecurity and can be used as a vehicle attack vector. Insufficiently protected diagnostic features could allow unapproved users to carry out chip tuning or repairs that don't adhere to technology standards, endangering the integrity or safety of the vehicle [6, 7].

In the UDS norm, special diagnostic services, namely service 0x27 (Security Access) and service 0x29 (Authentication), are defined to prevent an unauthorized test device from executing sensitive enhanced diagnostic services [8].

The UDS service 0x27, Security Access (SA), operates on the principle of a seed/key exchange mechanism. The communication between the vehicle ECU and diagnostic tester has the following flow (Figure 1):

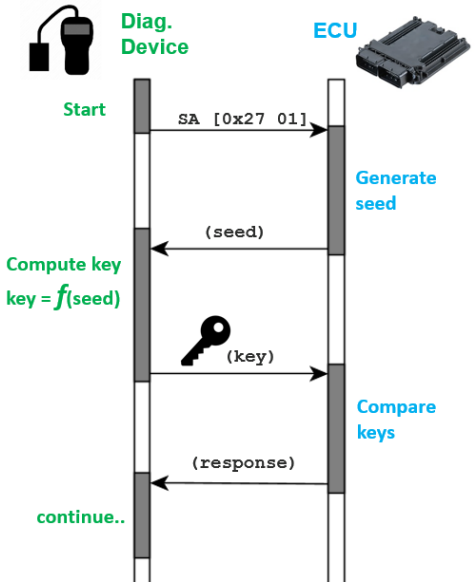


Fig. 1. UDS Security Access Service communication sequence diagram

1. The diagnostic tester initiates the SA procedure by sending the desired security level to an ECU.
2. The ECU responds by generating a random "seed" value (usually 32 bits). The ECU sends this seed value as a challenge to the diagnostic tool.
3. The diagnostic tool receives the seed value from the ECU and performs a calculation on it using a predetermined algorithm or function *f*(), typically based on a simple symmetric cryptographic method. This calculation generates a "key", which is sent back to the ECU. key = f(seed)
4. If the key generated by the diagnostic tool matches the key calculated by the ECU, temporary access for the ECU is granted, and the ECU sends a positive response to the diagnostic tester. If the keys are not equal, a negative response will be sent.

The UDS norm also specifies Service 0x29, Authentication, where the biggest difference (compared to Service 0x27) is the usage of asymmetric cryptography and certificates [8]. The communication between the vehicle ECU and diagnostic tester has the following flow during authentication (Figure 2):

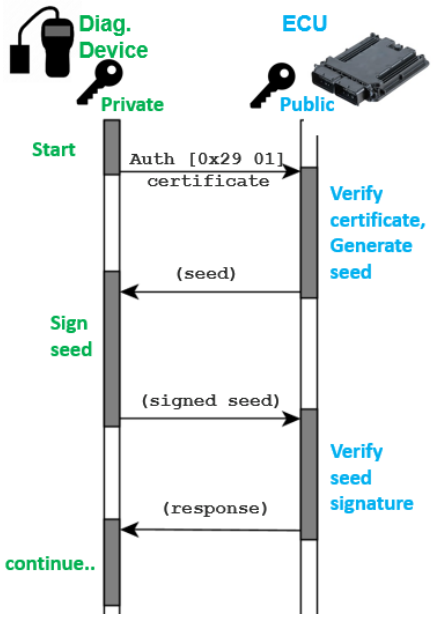


Fig. 2. UDS Authentication Service communication sequence diagram

1. The diagnostic tester initiates the authentication procedure by sending a diagnostic certificate to the ECU. The OEM private key signs the certificate.
2. First, the ECU verifies the certificate and its signature. To verify the received certificate, the ECU has its own public key (or certificate tree). The ECU responds by generating a random "challenge" value (256 bits or more).
3. The diagnostic tool is signing the received challenge using the private key corresponding to the public key included in the diagnostic certificate (proof of ownership). The diagnostic tool then sends the signed challenge back to the ECU.
4. If the valid diagnostic certificate on the ECU positively verifies the signed challenge, the ECU sends a positive response to the diagnostic tester, and the user (defined by the user or role of the diagnostic certificate) gains the authority to perform an advanced diagnostic functionalities.

A noteworthy difference between the mentioned services is that in the case of the SA service, the tester gains extended access within a diagnostic session consistent with the diagnostic session level.

Conversely, in the case of the 0x29 service, user authentication occurs, along with the assignment of specific privileges based on the user's role from the diagnostic certificate.

The vast majority of known and analysed scientific studies examining the influence of diagnostic protocols focus only on the standard legislative layer of OBD, excluding the subject of extended UDS protocols from their analyses [9, 10]. Some studies, however, proposing solutions to enhance cybersecurity suggest measures that impact the vehicle's electrical architecture (increasing costs) and fail to identify the source of the problem of the poorly secured UDS protocol [11]. This is likely because there are no official and documented sources of information on interpreting data from the OBD protocol, and the research results are not reproducible across different OEMs vehicles or controllers. This publication aims to expose the most fundamental errors in the implementation of UDS protocols that can be easily and inexpensively fixed during the software development phase and can be easily spotted during the automated software testing phase. One exception to this rule is the considerable problem of the lack of appropriate standards for diagnostic communication that could ensure full authentication of each diagnostic frame sent to the ECU. For instance, the well-established SecOC AUTOSAR standard only secures communication between controllers, not with the diagnostic tool, and there is currently no equivalent for diagnostic communication [12]. This problem is not easily solved and requires proprietary encryption solutions that, in practice, are not employed (as demonstrated in the article, a Man in the Middle (MITM) attack is relatively easy to execute).

1.2. Automotive security standards and regulations

The UDS norm is merely an exchange protocol standard and does not require the application of cybersecurity measures for diagnostic services at all. It provides only technical capabilities for its implementation. However, in recent years, two cybersecurity documents, UNECE R155 and ISO/SAE 21434, have been released to identify and assess the cybersecurity risks for motor vehicles. In 2021, the United Nations Economic Commission for Europe released the UN ECE-R155 regulation, which will be mandatory in some countries from 2024 to homologate vehicle cybersecurity. ISO/SAE 21434, officially released in 2021, is a cybersecurity standard that is intended to be widely applied in the engineering of electrical systems for road vehicles [13, 14]. These documents will have a significant impact on the security of onboard diagnostic systems, requiring vehicle manufacturers to pay more attention to the threats posed by diagnostic testers and external equipment connected to DLC port.

In the UN ECE-R155 for approval with the Cybersecurity Management System (CSMS) in the context of onboard diagnostics, the following vulnerabilities are recognized:

- "Threats to vehicles regarding their communication channels – Point 11.3 Malicious diagnostic messages"
- "Threats to vehicles regarding their external connectivity and connections – Point18.3 Diagnostic access (e.g. dongles in OBD port) used to facilitate an attack, e.g. manipulate vehicle parameters (directly or indirectly)"
- "Threats to vehicle data/code – Point 20.5 Unauthorized changes to system diagnostic data"

The regulation suggests the following mitigations:

- M10 "The vehicle shall verify the authenticity and integrity of messages it receives"
- M22 "Security controls shall be applied to external interfaces"
- M7 "Access control techniques and designs shall be applied to protect system data/code."

The defined mitigations have only relatively general and abstract meanings. Nevertheless, OEMs and TIER suppliers are responsible for determining whether, for instance, the diagnostic tester process should use symmetric or asymmetric authentication. Ideally, this decision should consider the Thread and Risk Analysis (TARA) evaluation, as suggested in ISO/SAE 21434.

Research results covered in later chapters demonstrate that onboard diagnostics remain inadequately secured and simple attacks can bypass current security measures as presented in the results.

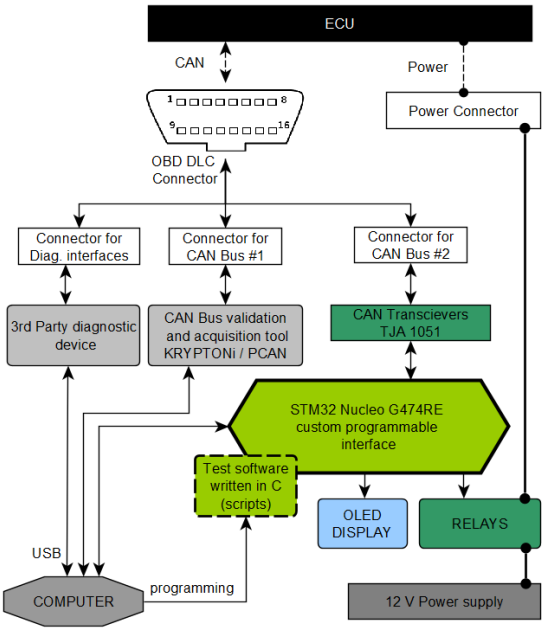
1.3 Test method and custom diagnostic device

The research was conducted on randomly selected motor vehicles and ECUs. All the experiments and testing were carried out through automated black box penetration tests (the authors had no access to the software or source code of the validated parts or vehicles). The tests were conducted on a custom-built diagnostic measuring station that allowed a direct connection to control units over a CAN Bus or connection to the vehicle over an OBD DLC port. The measuring station is a set of several different diagnostic interfaces and tools, but its most important element is custom interface built on STM32 Nucleo G474RE evaluation board [15]. The STM32 evaluation board includes a powerful STM32G474RE microcontroller that supports three independent CAN controllers allowing simultaneous communication on multiple buses [16]. Part of the custom diagnostic device in the prototyping phase is shown in Figure 3.

The custom-built diagnostic interface, thanks to the flexible programming in the C language, allowed for very efficient test implementation, execution, and result interpretation. The diagnostic interface was also equipped with an OLED display, multiple OBD connectors, a CAN data recorder, and a USB connection to the PC. In the case of a bench test of a

single ECU, it is also possible to control the ECU power supply.

Fig. 3. Diagram of custom-built measuring station for diagnostic interface testing



2. RESULTS

During the research, the following diagnostic interface security vulnerabilities were discovered.

2.1. Vulnerability 1: Executing UDS Command by the OBD Message

According to the norm, the OBD ECUs can be addressed by a common normalized CAN identifier - 0x07DF [5]. Because the normalized CAN identifier uses functional addressing, it will be received for processing by all the OBD-relevant ECUs. During the research, the authors discovered that one of the investigated vehicles was vulnerable to a simple attack, where a specially prepared diagnostic message was able to put a group of ECU's into error states. The special diagnostic message used the mentioned OBD standard (0x07DF) CAN Identifier, but the payload was manipulated; instead of an OBD PID, it uses a UDS service request - (0x02 10 03) Diagnostic Session Control – Extended Session. After sending the message to the vehicle over the OBD port, 10 ECUs processed it and replied with a positive response (0x50 03) (See Table 1). Because the UDS Extended Session stopped the ECU functions (the ECU went into special diagnostic mode), the vehicle was not operational anymore and the engine could not be started. This vulnerability is caused by an obvious software bug in the ECU firmware that allows for enhanced UDS diagnostic requests on a standard functionally addressed OBD CAN ID.

Table 1. Diagnostic CAN Communication trace, transmit a UDS Extended Diagnostic on standard OBD and the received positive responses from ECUs

time [s]	CAN ID	DIR.	Data
47.483	**07DF**	TX	02 **10 03**
49.559	07BB	RX	02 **50 03**
50.853	07D9	RX	06 **50 03** 00 32 01 F4
52.490	07CF	RX	06 **50 03** 00 32 01 F4
54.130	07DA	RX	06 **50 03** 00 32 01 F4
54.838	07CC	RX	06 **50 03** 00 32 01 F4
55.138	07DC	RX	02 **50 03**
58.601	07E8	RX	06 **50 03** 00 32 01 F4
61.040	0778	RX	06 **50 03** 00 32 01 F4
61.382	07A8	RX	06 **50 03** 00 32 01 F4
61.972	07CE	RX	02 **50 03** 0

2.2. Vulnerability 2: Seed value generation

During the Security Access procedure (Service 0x27), an ECU must generate a random number (seed), which is the input for the diagnostic tester for key calculation. To achieve an acceptable level of security, the seed value must demonstrate a suitable level of entropy. Serious security problems related to seed value generation were detected during the investigation across different ECUs.

The first identified vulnerability is related to the seed length. Despite the UDS prescribed seed value length of up to 4 bytes, controllers were found to generate values as short as 2 bytes (16 bits) (only 2^{16} possible seed values) (table 2).

Table 2. Diagnostic CAN Communication trace, requesting SA seed value and received responses

time [s]	CAN ID	DIR.	Data
3.720	18DA02F1	TX	02 10 03
3.727	18DAF102	RX	02 50 03
3.828	18DA02F1	TX	02 27 01
3.831	18DAF102	RX	04 67 01 **2C 64** 55 55
3.932	18DA02F1	TX	02 27 01
3.935	18DAF102	RX	04 67 01 **54 4D** 55 55

The potential brute force attack method becomes shorter due to the reduced seed value range and the ability to request 5 seeds per second, resulting in a success rate of no more than 4 hours (13107 seconds).

$$t = \frac{2^{16}}{5\ Hz} = 13107\ s \quad (1)$$

Carrying out an attack on this ECU is relatively simple, as it just needs repetitive readings of the seed value until a value corresponding to a known key is received.

Another interesting vulnerability related to questionable seed generation is the problem that, in some cases, the ECU returns the same and expected seed value. In one of the examined controllers, the random seed value remained constant when it was requested at the same point in time since the controller power-up. This reveals an undeniable software bug caused by the fact that the ECU has no access to a true random number generator and is probably generating the seed value based on the

internal timer, which is initialized with the same value every power-up cycle. The measurements of the ECU show that the internal timer/seed value is updated only every 8 ms, as shown on the chart in Figure 4.

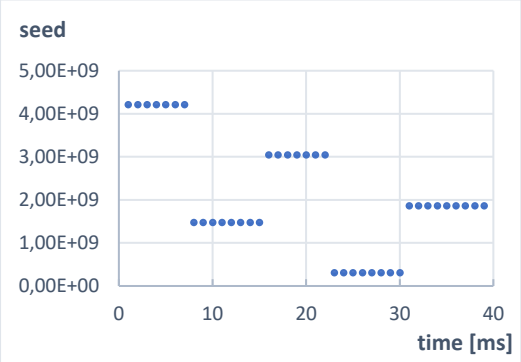


Fig. 4. Pseudorandom seed value read out after ECU Reset

In situations where a matching pair of seed and key is known (because it was logged on a different vehicle) and the time from power-up to seed read is known, it is possible to recreate the login procedure without knowing the secret cryptographic algorithm. An attack is relatively straightforward to conduct, as it only requires a specific attacker diagnostic tool capable of reading the seed value (initiate the SA service) at a defined point in time relative to power-up with an accuracy of 8 ms. This method will work on all ECUs of the same type assembled on different vehicles.

2.3. Vulnerability 3: Incorrect method of storing unsuccessful login attempts

Most of the inspected ECUs had an additional security mechanism that would lock the ability to perform SA service for a defined period in case the previous attempts failed because of the wrong key received. To implement this feature, the ECU must store in the non-volatile memory (NVM) an incremented counter value after an unsuccessful key comparison. During the research, it was observed that for one of the investigated ECUs, an additional delay time was needed to receive a negative response (when the key was incorrect). A positive response is received (correct key) in time t_1, and a negative response (wrong key) in time t_2, where there is always a relation $t_1 < t_2$. This behavior is very probably caused by the wrong (from a cybersecurity perspective) sequence of the software instructions, and the ECU requires additional time to store information inside the NVM before sending the negative response. However, it is important to mention that this behavior was observed only on two tested drivers, and the presented program sequence is only an assumption because the experiment was conducted in the form of black box testing. Figure 5 shows the supposed software sequence of operation of the vulnerable ECU.

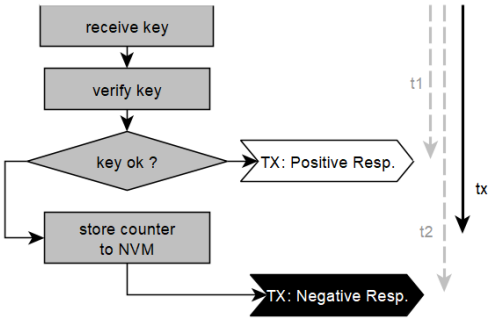


Fig. 5. Supposed ECU Software diagram of counting unsuccessful authorizations

The typical measured values are t_1 = 5 ms and t_2 = 12 ms. The difference of 7 ms between t_1 and t_2 was very probably caused by the fact that the microcontroller used in the ECU was using internal Flash memory, which is relatively slow (compared to RAM) and was causing additional latency in the program execution. This fact can be used against the ECU to perform a brute force attack in which the ECU's power supply is switched off at a precise point in time (t_x). Where $t_1 < t_x < t_2$. The test was executed on a test bench, and it was necessary to equip the attacking tool with a relay that could control the power supply of the investigated ECU as shown on Figure 6. The first tests were not successful because even when the diagnostic attacking device switched the power off in time t_x, the ECU was still able to run for some time, and it was still sending the negative response. The problem was caused by the internal electrical capacity of the ECU and could be solved by reducing the power supply V1 voltage from standard 12 V to 6,2 V.

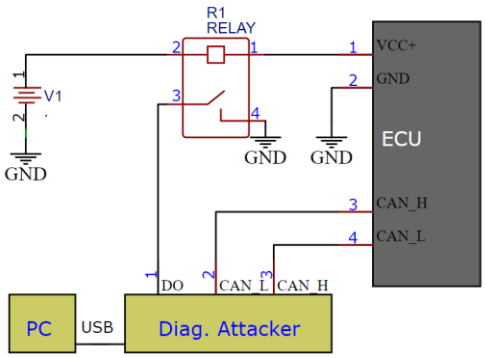


Fig. 6. Supposed ECU Software diagram of counting unsuccessful authorizations

Such a significantly reduced power supply voltage to the ECU still enabled its proper operation, but simultaneously caused its nearly instantaneous shutdown upon opening relay R1. As a result of the power break at t_x time, the diagnostic attacker device was able to perform a repetitive brute force attack by testing an unlimited number of wrong keys without locking the ECU. If no positive response was received in the time (t_x + 2 ms), relay R1 was opened. The ECU just did not have enough time to persist the error counter in NVM before power loss.

2.4. Vulnerability 4: Excessive unification of cryptographic algorithms

During the initial phase of the research on some ECUs, all attempts to breach the security of UDS service 0x27 were unsuccessful, which means the key for the Security Access service could not be calculated. However, in the later phase of the investigation, some knowledge acquired from other ECUs (which could be more easily exploited) of the same vehicle manufacturer could be utilized to infiltrate other ECUs. It turns out that the same secret cryptographic function *f*() used for UDS key calculation based on seed is reused between different ECUs types. For example, if the ECU controlling door of Vehicle_A is compromised (firmware read-out) and the function *f*() is known, it can be maybe used to unlock the engine ECU in Vehicle_B of the same manufacturer. This method applies only to some of the manufacturers and to ECUs that have approximately the same date of production. Unification of the function *f*() simplifies the diagnostic tools used by the workshops but leaves a clear security gap.

2.5. Vulnerability 5: Lack of diagnostic message authenticity

For ECUs that correctly implemented the UDS service Security Access 0x27 or used strong asymmetric cryptography in the service "Authenticity," there was no method found to overcome the security mechanisms of UDS locking. Despite the inability to calculate the correct UDS access key (or signature), the authors discovered an alternative method to attack the ECUs and inject their own diagnostic requests. This method utilizes the fact that when the official diagnostic tool establishes a connection with the ECU, the ECU will accept all diagnostic commands, regardless of the sender, until the diagnostic session is concluded. When a diagnostic tool has authenticated himself to an ECU, the ECU will open an authenticated session and remain in it until diagnostic communication is present (default timeout of 5 seconds). Because the UDS protocol (on the vehicle CAN bus) does not impose or even define the usage of full authentication for diagnostic communication, from the perspective of the receiver ECU, it is impossible to distinguish if the received commands are transmitted by the official authenticated toll or an attacking device. Currently, there is no diagnostic standard (for CAN buses), which would give the possibility of using full diagnostic message authenticity and integrity protection like the Transport Layer Security (TLS) used in computer networks [5, 8]. This situation makes all the ECUs using the UDS protocol on CAN bus (without applying not usual proprietary solutions) vulnerable to a man-in-the-middle (MITM) attack.

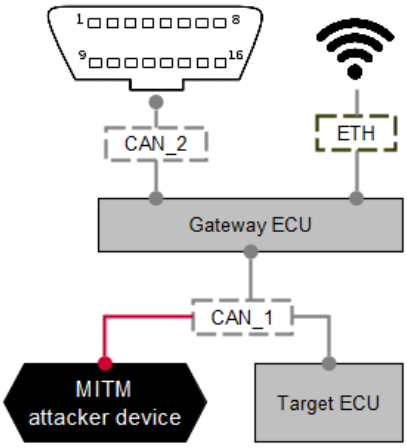


Fig. 7. Man in the middle attack connection topology

If the attacker aims to inject their own UDS commands (without authentication rights), it is enough to connect a properly prepared attacking device to the CAN bus of the selected ECU, which will wait and listen until the genuine diagnostic tool logs into the ECU. The schematic is shown on Figure 7. When the attacking MITM device detects the opening of an authenticated session, it might use the opportunity to inject its commands. This vulnerability may be used, for example, when a vehicle is getting a software update over-the-air, where a remote diagnostic device (indirectly) or a Gateway ECU (directly) is authenticating to the updated ECU. After the successful authentication of the authorized remote diagnostic tool, some additional UDS messages can be injected (like a new calibration request) by the nonauthorized MITM tool. During our tests on the example vehicle, the target ECU was not able to distinguish if the received diagnostic requests were sent by the Gateway ECU or MITM device.

This vulnerability is eliminated only on systems using the Ethernet ECU communication and DoIP (Diagnostic over Internet Protocol) protocols, which, unfortunately, because of their high cost, are used only in premium vehicles [17, 18]. There is also no SecOC (Secure Onboard Communication) or any similar solution that would secure the external communication to diagnostic devices [12, 20].

3. RESULTING SECURITY RISK MITIGATION REQUIREMENTS FOR DIAGNOSTIC

Appropriate software patches in the ECUs firmware can partially or entirely mitigate most of the aforementioned cybersecurity threats related to UDS diagnostics. The authors would propose to add the following software requirements for UDS security implementation:

I. UDS-specific services shall not be accessible by the standard OBD CAN identifier diagnostic messages defined in the ISO norm [5]. UDS Extended diagnostic messages shall employ

distinct CAN identifiers, permitting processing only under specific, approved circumstances.

II. The implementation of UDS Security Access Service (0x27) shall use a true random number generator (TRNG) for seed generation. If the TRNG generator is not available in the hardware, it must be assured that the pseudo-random number generator (PRNG):
   i. Every power-up cycle is initialized with a value from the previous power cycle. Non-volatile memory shall be used to persist the previous PRNG state.
   ii. All PRNG inputs must have an update rate frequency that is higher than 10/CBT (CBT-CAN Bit Time).
   iii. In any ECU condition or state, none of the inputs shall represent a constant value.
   iv. The output shall use a 64-bit value or higher.

III. Consecutive unsuccessful SA or "Authentication" attempts shall be counted and stored in non-volatile memory. The attempt counter shall be incremented when the seed is generated, not when the wrong key is received. Once the ECU reaches a certain number of unsuccessful attempts, it shall not accept any new requests until a minimum delay time has elapsed.

IV. After receiving a key or signature, the ECU response shall be sent asynchronously, independent of the correctness of the key or signature. If needed, additional delay shall be added before sending a positive response. A positive or negative response shall be sent only if all data has persisted in ECU memory.

V. The key generation algorithm for the service SA shall take unique ECU/vehicle identification data as input (for example, SW Version or Vehicle VIN Number). This will prevent the reuse of a known seed and key pair on a different vehicle or ECU.

VI. When the SA service is used, distinct security levels with differing algorithms must be implemented to authorize access to diagnostic functionalities of different types. For example, a distinct key (security level) shall be used to enable uploading new software than for actuator tests.

VII. When the "Authentication" service is used, distinct security certificates with differing keys must be verified to authorize access to diagnostic functionalities of different types. For instance, uploading new software should be enabled using a different certificate (security role) than changing encoding.

VIII. An additional UDS diagnostic routine shall be implemented, which will end the authenticated/security diagnostic session. This routine shall be used by the diagnostic tool to immediately cancel the SA or "Authentication" after finishing all diagnostic procedures (without waiting for any timeouts).

4. CONCLUSIONS

Although the automotive sector has adopted appropriate standards and legal regulations mandating cybersecurity measures, software glitches in controllers still create security vulnerabilities in the diagnostic systems. Despite the potential of on-board diagnostic systems (UDS) to significantly influence automotive vehicles and modify their technical parameters, most investigated vehicles or ECUs appear to lack adequate protection against cyberattacks. The authors of the article, using their own measuring station and proprietary software for verifying diagnostic functions, detected several critical gaps in the controller software. The research confirmed that it is possible to hack into controllers through their diagnostic interface without incurring significant costs and having no access to private functions or cryptographic keys. Especially the UDS services "Security Access" and "Authenticity" are vulnerable to brute force and replay attacks. Suggested software fixes and proper verification could easily eliminate most of the exposed vulnerabilities by the proposed additional requirements or OWASP verification standard [21]. The authors proposed a set of software requirements that can improve the security of diagnostic functions when implemented correctly.

A significant cybersecurity threat to diagnostic systems stems from the reliance on the CAN bus and the UDS protocol, which do not define a standard for encrypting or authenticating diagnostic communication between the vehicle and external test equipment. Vehicle manufacturers and ECU suppliers must intensify their efforts in the software development process to enhance the cybersecurity of their products.

Source of funding: *Research work 12/010/BK_24/1151 Department of Road Transport Faculty of Transport and Aviation Engineering Silesian University of Technology.*

Author contributions: *research concept and design, P.P., Ł.K.; Collection and/or assembly of data, P.P.; Data analysis and interpretation, P.P.; Writing the article, P.P.; Critical revision of the article, Ł.K.; Final approval of the article, Ł.K.*

Declaration of competing interest: *The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.*

REFERENCES

1. ISO 15031-5:2015 Road vehicles - Communication between vehicle and external equipment for emissions-related diagnostics. Part 5: Emissions-related diagnostic services.
2. ISO 15031-3:2023 Road vehicles - Communication between vehicle and external equipment for emissions-related diagnostics. Part 3: Diagnostic

connector and related electrical circuits: Specification and use.
3. Witaszek K, Witaszek M. Diagnosing the thermostat using vehicle on-board diagnostic (OBD) data. Diagnostyka. 2023;24(4):2023402. https://doi.org/10.29354/diag/173002.
4. Wierzbicki S. Evaluation of the effectiveness of on-board diagnostic systems in controlling exhaust gas emissions from motor vehicles. Diagnostyka. 2019;20(4):75-79. https://doi.org/10.29354/diag/114834
5. ISO 15765-4:2021 Road vehicles - Diagnostic communication over Controller Area Network (DoCAN). Part 4: Requirements for emissions-related systems.
6. Bozdal M, Samie M, Aslam S, Jennions I. Evaluation of CAN Bus Security Challenges. Sensors 2020; 20: 2364. https://doi.org/10.3390/s20082364
7. Luo A, Spencer H. Remotely hacking a car through an OBD-II Bluetooth Dongle [Video]. Youtube. https://www.youtube.com/watch?v=f19_BNgVrWQ&ab_channel=AutomotiveSecurityResearchGroup
8. ISO 14229-1:2020 Road vehicles - Unified diagnostic services (UDS) — Part 1: Application layer.
9. Kim H, Jeong Y, Choi W, Lee DH, Jo HJ. Efficient ECU analysis technology through structure-aware CAN fuzzing. IEEE Access, 2022;10:23259-23271. https://doi.org/10.1109/ACCESS.2022.3151358
10. Kang TU, Song HM, Jeong S and Kim HK. Automated reverse engineering and attack for CAN using OBD-II. 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, USA, 2018:1-7. https://doi.org/10.1109/VTCFall.2018.8690781
11. Ammar M, Janjua H, Thangarajan A, Crispo B. Securing the On-Board Diagnostics Port (OBD-II) in vehicles. SAE International Journal of Transportation Cybersecurity and Privacy *2019;* 2(2):83-106, 2019. https://doi.org/10.4271/11-02-02-0009.
12. AUTOSAR Group. Specification of secure onboard communication protocol (SecOC) R21-11, 2021. https://www.autosar.org
13. ISO/SAE 21434:2021 Road vehicles - Cybersecurity engineering.
14. UN Regulation No. 155 - Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system, 2021.
15. STMicroelectronic. User Manual. UM2505 STM32G4 Nucleo-64 boards (MB1367). https://www.st.com/resource/en/user_manual/um2505-stm32g4-nucleo64-boards-mb1367-stmicroelectronics.pdf
16. STMicroelectronic. Reference manual RM0440 STM32G4 series advanced Arm®-based 32-bit MCUs https://www.st.com/resource/en/reference_manual/rm0440-stm32g4-series-advanced-armbased-32bit-mcus-stmicroelectronics.pdf
17. Matsubayashi M, Koyama T, Tanaka M. In-Vehicle network inspector utilizing diagnostic communications and web scraping for estimating ECU functions and CAN Topology. IEEE Access 20214; 12: 6239-6250. https://doi.org/10.1109/ACCESS.2024.3351175.
18. Ajin VW, Kumar LD, Joy J. Study of security and effectiveness of DoIP in vehicle networks. 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 2016; 1-6. https://doi.org/10.1109/ICCPCT.2016.7530357.
19. Śmieja M, Wierzbicki S, Mamala J. Sterowanie dawką wtryskiwanego paliwa w układzie Common Rail z wykorzystaniem środowiska LabView. Combustion Engines. 2013;154(3):542-548.
20. Mokhadder M, Zachos M, Potter J. Evaluation of Vehicle System Performance of an SAE J1939-91C Network Security Implementation (2023) SAE Technical Papers. https://doi.org/10.4271/2023-01-0041
21. OWASP Application Security Verification Standard, 10 2020, [online] Available: https://owasp.org/www-project-application-security-verification-standard/

Piotr PEŁECHATY received M.Sc. degree in Department of Road Transport, Faculty of Transport and Aviation Engineering, Silesian University of Technology. He works as a software engineer in the automotive sector, developing cybersecurity solutions for embedded systems. He is conducting research for the implementation PhD.
e-mail: piotr.pelechaty@polsl.pl

Łukasz KONIECZNY is DSc PhD in Department of Road Transport, Faculty of Transport and Aviation Engineering, Silesian University of Technology. His research interests are: machinery vibrodiagnostic, digital analyse of signals, simulation researches of vehicle suspension dynamics, hydro-pneumatic suspensions.
e-mail: lukasz.konieczny@polsl.pl