Intrusion Detection System for SOME/IP-Based In-Vehicle Network

Subjects: Engineering, Electrical & Electronic

Contributor: Feng Luo , Zhenyu Yang , Zhaojing Zhang , Zitong Wang , Bowen Wang , Mingzhi Wu

This research describes a hybrid AI and rule-based intrusion detection system for detecting SOME/IP communication threats in new automotive electronic electrical architectures.

intrusion detection system SOME/IP deep learning

1. Introduction

With the continuous evolution of the Internet of Things (IoT), the vehicle has become an indispensable part [1]. The trend of IoT leads to the introduction of information technology (IT), software-defined networking (SDN) [2], and service-oriented architectural design concepts, which give automotive applications great flexibility to deploy, update, and expand the introduction of information technology (IT), and service-based architectural design concepts give automotive applications great flexibility to deploy, update and expand [3]. A large amount of external data enters the IVN through wireless technologies, such as Wi-Fi, Bluetooth, ZigBee, dedicated short-range communication (DSRC), and long-term evolution (LTE). Diverse upper-layer applications, such as safety-related, entertainment-related, and control-related applpications [4], also put forward new requirements for the backbone of the IVN. In addition to high speed and high bandwidth, the in-vehicle network also needs to be redundant, scalable, real-time, deterministic, and secure, which cannot be provided by traditional in-vehicle buses such as CAN, local interconnect network (LIN) and media-oriented system transport (MOST). The automotive Ethernet (AE) solves the electromagnetic compatibility problem using traditional Ethernet in the vehicle environment [5]. The above requirements can be satisfied by optimizing and multiplexing the protocols of different layers in the OSI model to the AE [6]. BMW proposed the SOME/IP protocol in 2011 as a critical protocol for solving service-oriented communication and was incorporated into the AUTomotive Open System Architecture (AUTOSAR) specification in 2014. Kreissl [7] obtained the vulnerability of SOME/IP through threat analysis, and some studies have proven that the in-vehicle network can be hacked through external interfaces (Bluetooth, WIFI), operating system vulnerabilities, or malware [8][9][10][11]. It is evident that SOME/IP has security risks and needs corresponding security measures.

2. Vulnerability of SOME/IP

SOME/IP is built on the TCP/UDP protocol and is located above the fourth layer of the OSI model. Its purpose is to define a unified middleware for IP-based communication within the vehicle. SOME/IP is one of the critical components to realizing the in-vehicle network communication under the service-based architecture.

2.1. SOME/IP Overview

The communication based on SOME/IP is divided into two phases. The first is the service discovery process, specified by SOME/IP Service Discovery Protocol [12], and the second is the normal communication process, specified by SOME/IP Protocol [13]. The SOME/IP-SD message and SOME/IP message format are shown in **Figure 1**.

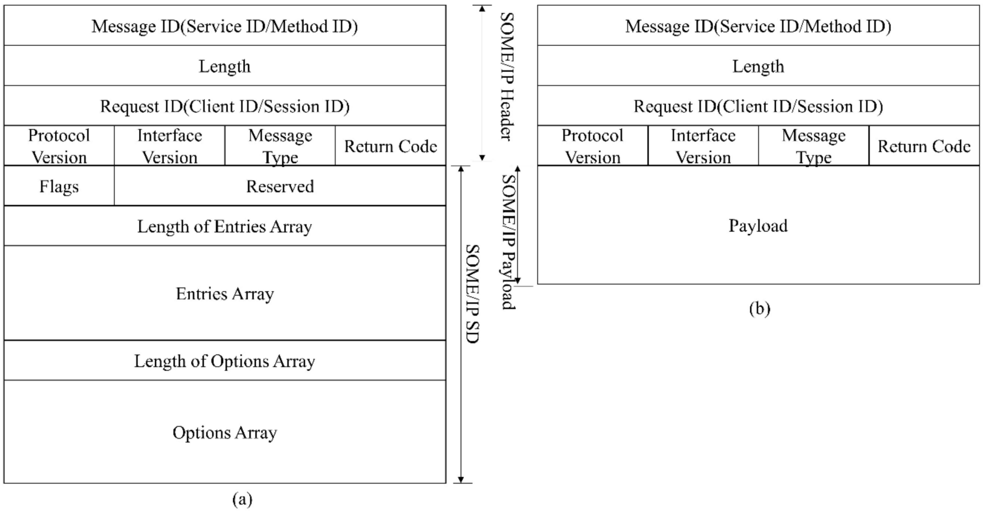

Figure 1. SOME/IP protocol format. (**a**) SOME/IP-SD message format. (**b**) SOME/IP message format.

The service discovery process is performed when the system starts, including three phases: initial wait, repetition, and main. Servers and clients notify each other of service information through SOME/IP-SD messages, consisting of the entries array and options array.

A service consists of combinations of zero or multiple events, methods, and fields. Events provide data sent cyclically or on change from the provider to the subscriber. A field does represent the status and thus has a valid value at all times upon which the getter, setter, and notifier act. The communication of SOME/IP relies on RPC and Publish-Subscribe. RPC allows the client to call methods in the server. RPC contains two modes, Fire & Forget and Request-Response. The difference is that Fire & Forget does not need a response. Events in the service can only be transmitted after they have been subscribed. Operations of the field are special since the setter and getter of the field belong to Request-Response-RPC, while the notifier of the Field needs to be subscribed like an event. The communication paradigm of SOME/IP is shown in **Figure 2**.

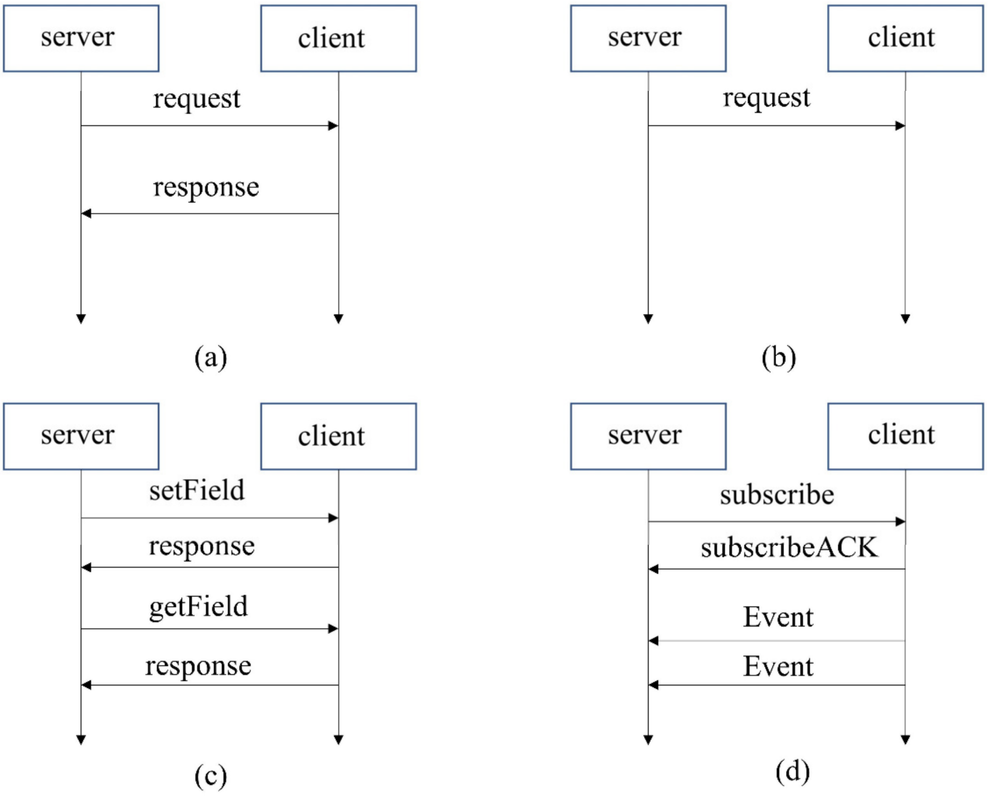

Figure 2. SOME/IP communication paradigm. (**a**) Request-Response-RPC. (**b**) Fire & Forget-RPC. (**c**) Setter & Getter of Field. (**d**) Publish-Subscribe for event.

In SOME/IP communication, events and RPC often act on different types of vehicle data. Since autonomous driving control algorithms require periodic and continuous inputs, the event is more suitable for real-time control and is primarily a fundamental signal. If the self-driving application uses RPC to trigger related calculation signals, it will increase the network load and reduce the real-time performance, resulting in a poor control effect. In contrast, RPC is more suitable for the interaction between humans and the vehicle or the control of body parts with low real-time requirements, such as calling the air conditioning control method or the turn signal control method through RPC.

2.2. Attack Scenario

Figure 3 shows a zonal automotive electrical and electronic architecture (EEA). The automotive Ethernet is used as the backbone network to connect the zonal control unit (ZCU), central compute unit (CCU), rear seat entertainment (RSE), and telematics box (T-Box) in a star topology. SOME/IP runs as an upper-layer protocol in the backbone network. Four ZCUs are in charge of each of the four zones in the left, right, front, and rear of the vehicle. Each zone’s actuators, sensors, and sub-ECUs are connected to the ZCU via CAN or Ethernet. There are various external interfaces including Bluetooth, cellular network, GPS in RSE, CCU, and T-Box. Due to the need to perform diverse tasks such as information fusion, route planning, infotainment, etc., these electronic units are equipped with diverse operating systems such as Android, QNX, and Linux. An attack from the outside to the in-vehicle SOME/IP network is possible under the above EEA.

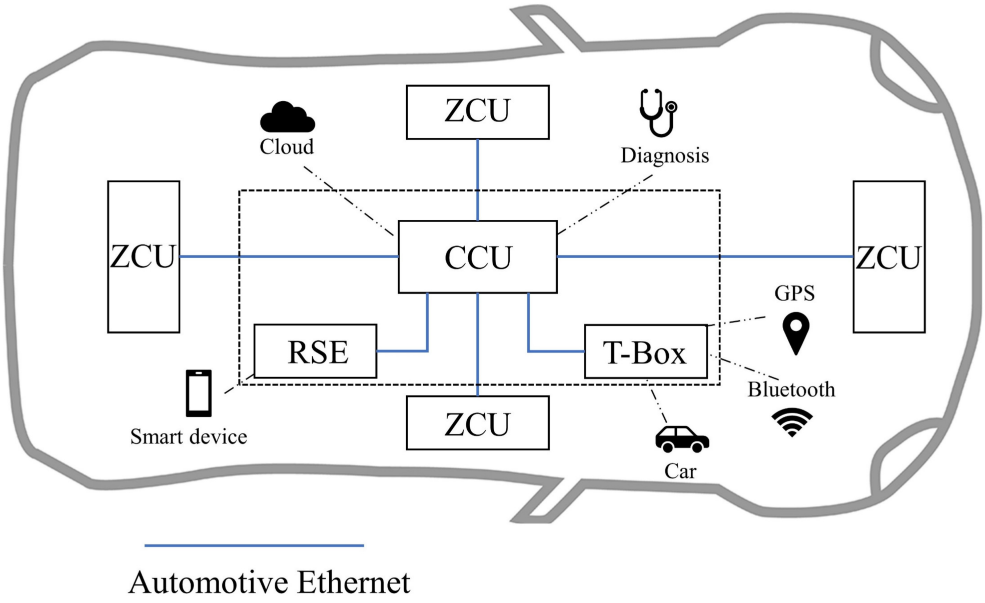

Figure 3. Automotive zonal EEA.

Due to the fixed topology of the AE-based IVN and the point-to-point communication method, it is almost impossible to attack the in-vehicle network by mounting malicious communication nodes directly, except in the ideal case. However, an attacker can infiltrate the SOME/IP network outside. A more feasible approach is that an attacker attacks straight from the data source, such as spoofing the camera, causing the speed sensor to have measurement errors, etc. In this scenario, all ECU nodes in the IVN are normal and communicate as expected. Moreover, there may be vulnerabilities in applications, operating systems, or virtual machines. It is possible to gain access to data or the network through these vulnerabilities or malicious software. For example, when an attacker obtains permission to operate the transmit interface of SOME/IP packet, attacks such as replay, tamper, fuzzy, and denial of service (DoS) can be launched on the network.

2.3. Attack of SOME/IP

2.3.1. Fuzzy

Targets of fuzzy include the header of the event and RPC, service entries array, and option array in the service discovery packet. Fuzzy can also be understood as random or traversal tampering.

2.3.2. Spoof

Spoof is considered an upgrade of Fuzzy. In the definition, the targets of Fuzzy do not contain the payload. The Fuzzy on the payload is invalid if the SOME/IP header does not match the requirements. Spoof means that the attacker can send the header of SOME/IP as required and tamper or replay the payload of the event at the same time. This requires a higher level of mastery of communication systems.

2.3.3. DoS

DoS refers to the attacker congesting the network by modifying the cycle of periodic Events or SOME/IP-SD packets. DoS can also be achieved by injecting large amounts of traffic unrelated to SOME/IP. Nevertheless, this is not a SOME/IP level attack, and researchers can solve such DoS from the data link layer by introducing a flow meter or IEEE 802.1Qci [14].

2.3.4. Abnormal Communication Process

The abnormal communication process mainly involves the four steps mentioned in the research [15], including error on error, error on event, missing response, and missing request.

2.3.5. Unauthorized Operation

Unauthorized operations do not exist in conventional CAN buses, which is mainly manifested in unauthorized subscription, unsubscription, provision of services, and unauthorized RPC calls. The services or RPCs here are defined in the system but have not been authorized by the upper application.

3. Proposed Multi-Layer IDS

3.1. Dataset Generation

So far, there is no recognized SOME/IP dataset for intrusion detection in the industry. Since SOME/IP-based service-oriented communication has not been widely deployed in mass-produced vehicles, actual vehicle data communicating via SOME/IP cannot be collected. The literature [16] provides a SOME/IP data generator, but this generator can only generate the header of SOME/IP and fill in the payload with random numbers or fixed values. This makes it inconvenient to conduct a comprehensive intrusion detection study. The toolchain of Prescan, Simulink, and Vector CANoe is used to build the SOME/IP dataset to fill this gap. Prescan is a simulation platform for ADAS function development, which integrates modules involved in intelligent driving simulation, such as road scenes, smart cars, sensor models, vehicle dynamics configuration, and environmental perception. Simulink is a block diagram environment for modeling, analyzing, and simulating dynamic systems. CANoe is a bus development environment produced by VECTOR, which can be used for modeling, simulation, testing, and development of automotive buses. Simulating the ADAS function with Prescan and Simulink is one of the most common methods nowadays. Moreover, CANoe produced by Vector is widely used for in-vehicle network simulation and testing. Through this toolchain, traffic that meets the protocol requirements and has ADAS meaning can be generated. The data generation process is shown in **Figure 4**.

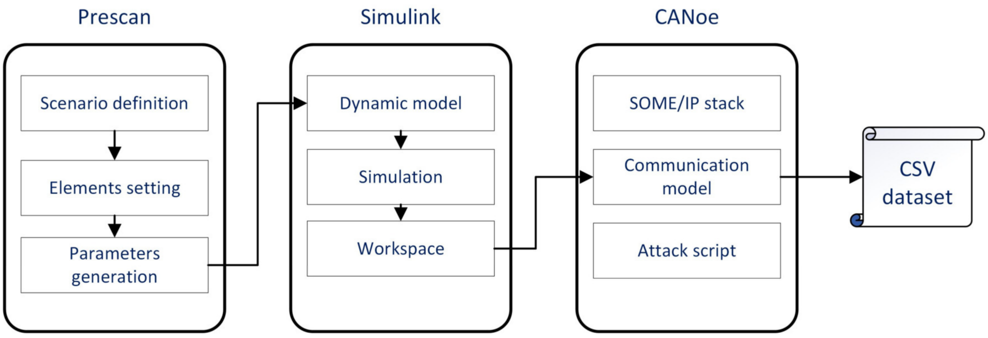

Figure 4. SOME/IP dataset generation process for IDS.

Prescan is first used to design and build vehicle simulation scenarios. Road elements such as vehicles and road signs can be added according to actual needs. A detailed set of parameters constrains each element. For example, the weight, wind resistance, running trajectory, dynamic parameters, etc., can be set for vehicle elements. The simulation scenarios and parameters defined in Prescan are then imported into the vehicle dynamics model and application sub-functions built in Simulink. Interrelated data in the simulation environment can be obtained, such as the sensor data, vehicle speed, throttle opening, hydraulic braking force, etc. The above data are then imported into CANoe and encapsulated by the SOME/IP protocol stacks implemented by the communication access programming language (CAPL). According to the defined service framework, these SOME/IP messages are transmitted between simulation nodes in CANoe. Finally, these messages are recorded through the logging file and form a CSV database through Python. The attack script is coded with CAPL and embedded in the emulation node. The attack can be executed via the panel, similar to the attack through the APP backdoor. It should be noted that this is not an actual attack scenario, but the same attack effect can be obtained.

3.2. System Structure

The multi-layer IDS consists of rule-based detection and AI-based detection. The models in AI-based detection are trained before the IDS works appropriately. **Figure 5** describes the system architecture of multi-layer IDS and reflects its workflow. In the training phase, the data are imported from the database into the data pre-processing module, which includes feature deserialization, normalization, and sequence generation. Sequence enters the initial multi-GRU model for training, and Bayesian optimization is used to obtain the hyperparameters of the model. During the detection phase, real-time SOME/IP packets flow from the IVN to the IDS. The packet enters the data extraction module, where the features for IDS are extracted. These features first enter the rule-based intrusion detection module. When all the rules are passed, the event packet will go through the pre-processing module to generate the feature sequence and enter the AI detection module. Other types of packets will jump out directly after entering the AI detection module and are marked as normal. When the results of both detection modules are normal, the packet is classified as normal. If any rules are not satisfied or the detection result of AI-based IDS is abnormal, the IPS will trigger related protection mechanisms, such as alerting, isolation, etc.

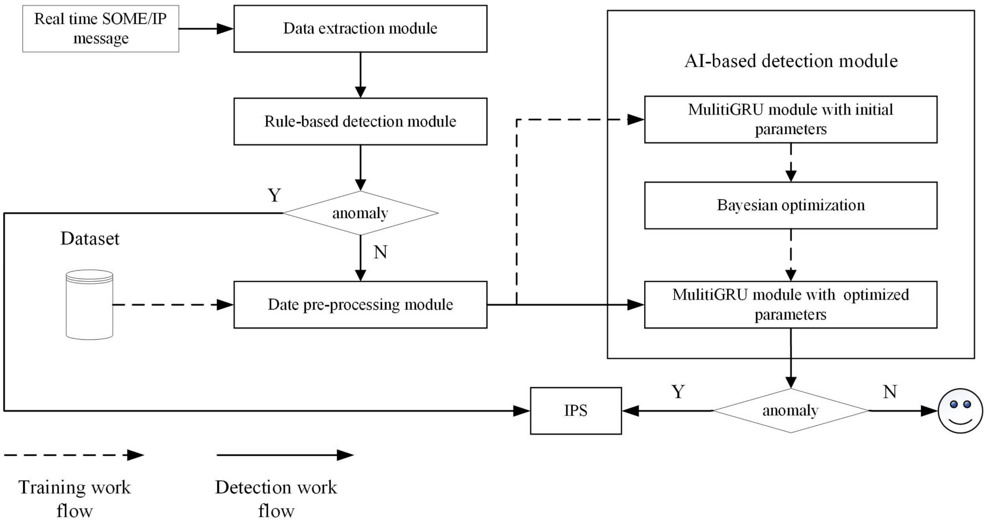
Real time SOME/IP message
Data extraction module
Rule-based detection module
anomaly
Y
N
Dataset
Date pre-processing module
AI-based detection module
MulitiGRU module with initial parameters
Bayesian optimization
MulitiGRU module with optimized parameters
IPS
anomaly
Y
N
Training work flow
Detection work flow

Figure 5. System architecture and workflow of multi-layer IDS.

The detection range of the multi-layer IDS list is outlined in **Table 1**. For example, the driver can control the radio volume after the vehicle ignition, in which case an authorized RPC is generated. However, the attacker can issue an unauthorized RPC to control the radio volume anytime. The context of these two messages in the network is irregular. Such an attack cannot be identified at the network level if the service encapsulating RPC is offered. Hence, NIDS is only one part of the defense-in-depth system. Unauthorized operations must be defended by application probes, access control, or a host-based intrusion detection system (HIDS).

Table 1. The detection range of multi-layer IDS.

Attack Type	Detection Module	Content
Fuzzy	Rule-based	Header of event, RPC and SOME/IP-SD packet; service entries array and options array in SOME/IP-SD packet
Spoof	AI-based	Payload of SOME/IP event
DoS	Rule-based	Interval of event and SOME/IP-SD packet
Abnormal communication process	Rule-based	Communication process

3.3. Data Extraction Module

Data extraction is a layer-by-layer unpacking process necessary for Ethernet communication. Each time the initial SOME/IP packet passes through a layer of the OSI model, the header of that layer will be added. After unpacking, the features for detection can be obtained.

3.4. Rule-Based Detection Module

All SOME/IP packets are first subjected to rule-based detection, which can detect anomalies in the SOME/IP-SD packet, SOME/IP header of event and RPC, and communication process. Each message ID, consisting of service ID and method ID, as shown in **Figure 1**, corresponds to a rule group. The rule group includes static field rules, dynamic field rules, and communication state rules. These rules will be judged in turn. As soon as a rule is not satisfied, an anomaly is flagged and pops up immediately. The packet is also marked as an anomaly if there is no corresponding message ID in the module. Parameters in the whole rule-based detection module are listed in **Table 2**.

Table 2. Fields used in rule-based detection module.

Field Name	Description
IP Address	Static field in the header of SOME/IP and SOME/IP-SD packet
MAC address	
Port number	
Message ID	
Protocol version	
Interface version	
Message type	
Client ID	
Find service entries array	
Offer service entries array	
Eventgroup array	
Options array	
Session ID	Dynamic field
Interval of packet	
Status cache of the previous message	Communication status parameter
The estimated status of the next message	

The static fields mainly include IP Address, MAC address, port number, message ID, protocol version, interface version, message type, and information in the entries array and options array. These fields and their matching relationships are fixed after completing the service architecture and network topology design, such as the services the nodes can provide, the methods or event groups contained in the services, etc. Service description files can be extended through software-over-the-air (SOTA). In this case, static rules also need to be updated synchronously.

Dynamic fields refer to the timestamp of the SOME/IP packet and session ID in the SOME/IP header. The growth logic of session ID is defined in the specification [13] in detail. Attacks not complying with the session ID growth rules can be easily detected by checking this field, such as replay, tamper, and injection without the correct session id. Since SD and event packets are sent periodically, the injection of periodic packets can be detected by comparing the frame interval calculated by the timestamp with the set threshold.

The communication status rule detects the four process errors defined in the paper [15], error on error, error on event, missing response, and missing request. The rule-based module will cache the information of the previous packet. For example, after receiving a response-type SOME/IP packet, the module will check whether the previous frame of the same message ID is the request type.

References

1. Keertikumar, M.; Shubham, M.; Banakar, R.M. Evolution of IoT in smart vehicles: An overview. In Proceedings of the International Conference on Green Computing and Internet of Things (ICGCIoT 2015), Greater Noida, India, 8–10 October 2015; pp. 804–805.
2. Toufga, S.; Abdellatif, S.; Assouane, H.T.; Owezarski, P.; Villemur, T. Towards Dynamic Controller Placement in Software Defined Vehicular Networks. Sensors 2020, 20, 1701.
3. Traub, M.; Maier, A.; Barbehon, K.L. Future Automotive Architecture and the Impact of IT Trends. IEEE Softw. 2017, 34, 27–32.
4. Panigrahy, S.K.; Emany, H. A Survey and Tutorial on Network Optimization for Intelligent Transport System Using the Internet of Vehicles. Sensors 2023, 23, 555.
5. Hank, P.; Müller, S.; Vermesan, O.; Van Den Keybus, J. Automotive ethernet: In-vehicle networking and smart mobility. In Proceedings of the 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE 2013), Grenoble, France, 18–22 March 2013; pp. 1735–1739.
6. Aspestrand, O.; Claeson, V. The Fast-Lane Development of Automotive Ethernet for Autonomous Drive. Master’s Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2018. Available online: https://odr.chalmers.se/bitstream/20.500.12380/256211/1/256211.pdf (accessed on 20 April 2023).
7. Kreissl, J. Absicherung der SOME/IP Kommunikation Bei Adaptive AUTOSAR. Master’s Thesis, Universität Stuttgart, Stuttgart, Germany, 2017. Available online: https://elib.uni-stuttgart.de/bitstream/11682/9482/1/ausarbeitung.pdf (accessed on 20 April 2023).
8. Golson, J. Jeep Hackers at It Again, This Time Taking Control of Steering and Braking Systems. Available online: https://www.theverge.com/2016/8/2/12353186/car-hackjeep-cherokee-vulnerability-miller-valasek (accessed on 20 April 2023).

9. Miller, C.; Valasek, C. A Survey of Remote Automotive Attack Surfaces. Available online: https://img.hardworkingtrucks.com/files/base/randallreilly/all/migrated-files/hwt/2014/09/Remote_Automotive_Attack_Surfaces.pdf (accessed on 20 April 2023).
10. Woo, S.; Jo, H.J.; Lee, D.H. A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN. IEEE Trans. Intell. Transp. Syst. 2014, 16, 993–1006.
11. Mandal, A.K.; Cortesi, A.; Ferrara, P.; Panarotto, F.; Spoto, F. Vulnerability analysis of Android auto infotainment apps. In Proceedings of the 15th ACM International Conference on Computing Frontiers, Ischia, Italy, 8–10 May 2018; pp. 183–190.
12. SOME/IP Service Discovery Protocol Specification, AUTOSAR. 2022. Available online: https://www.autosar.org/fileadmin/standards/R22-11/FO/AUTOSAR_PRS_SOMEIPServiceDiscoveryProtocol.pdf (accessed on 20 April 2023).
13. SOME/IP Protocol Specification, AUTOSAR. 2022. Available online: https://www.autosar.org/fileadmin/standards/R22-11/FO/AUTOSAR_PRS_SOMEIPProtocol.pdf (accessed on 20 April 2023).
14. Luo, F.; Wang, B.; Fang, Z.; Yang, Z.; Jiang, Y. Security Analysis of the TSN Backbone Architecture and Anomaly Detection System Design Based on IEEE 802.1Qci. Secur. Commun. Netw. 2021, 2021, 6902138.
15. Alkhatib, N.; Ghauch, H.; Danger, J.-L. SOME/IP Intrusion Detection using Deep Learning-based Sequential Models in Automotive Ethernet Networks. In Proceedings of the 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON 2021), Vancouver, BC, Canada, 27–30 October 2021; pp. 0954–0962.
16. SOME/IP Generator. Available online: https://github.com/Egomania/SOME-IP_Generator (accessed on 20 April 2023).

Retrieved from https://encyclopedia.pub/entry/history/show/101126