CHALMERS
UNIVERSITY OF TECHNOLOGY

Indicators of Compromise of Vehicular Systems

Master's thesis in Computer Systems and Networks

Mohammad Hadi Sultani & Lu Han

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Indicators of Compromise of Vehicular Systems

Mohammad Hadi Sultani & Lu Han

Department of Computer Science and Engineering
Division of Networks and Systems
Chalmers University of Technology
Gothenburg, Sweden 2019

Indicators of Compromise of Vehicular Systems
Mohammad Hadi Sultani & Lu Han

Supervisor: Tomas Olovsson, Department of Computer Science and Engineering
Advisors: Christian Sandberg and Manne Engelke, Volvo Group Trucks Technology
Examiner: Tomas Olovsson, Department of Computer Science and Engineering

Master's Thesis 2019
Department of Computer Science and Engineering
Division of Networks and Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Volvo FH truck. © Volvo Truck Corporation. All rights reserved.

Typeset in LaTeX
Gothenburg, Sweden 2019

Indicators of Compromise of Vehicular Systems

Mohammad Hadi Sultani & Lu Han
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Modern vehicles are no longer mere mechanical devices; they are equipped with plenty of sensors and Electronic Control Units (ECUs) for their primary functions such as powertrain and brake systems. Some legislation mandates the use of ECUs in the modern vehicles because the pure mechanical solutions such as legacy carburetors or hydraulic brake systems can neither comply with the safety and emission regulations nor achieve the consumers' demands. The number of ECUs in most modern vehicles goes beyond one hundred. To achieve higher consumer satisfaction, vehicle manufacturers also implement plenty of built-in advanced entertainment and navigation systems which in most cases require an Internet connection.

By connecting to the Internet, to other vehicles, and to infrastructures, as well as having hundred of millions of lines of code, vehicles have emerged as drivable computers. Similar to ordinary computers, modern vehicles are also exposed to different types of cyber-attacks which can cause safety issues for the driver, the passengers, and other properties.

Nonetheless, there has been much research within this area; especially on Intrusion Detection Systems (IDS). However, there are still some issues with the IDSs, and the most significant one is the high rate of false alarms considering the massive number of vehicles deployed in the market.

In this thesis project, we introduced many Indicators of Compromise (IOC) in vehicular systems. Indicators of Compromise are simple artifacts whose presence in a system is a sign of intrusion or infection by malicious software. The IOCs trigger if the legitimate behavior of the system is violated; thus can mitigate the number of false positives if implemented and deployed on the system. Also, we have defined a set of criteria and methodologies in order to conduct a qualitative evaluation of the IOCs in order to determine their quality. Additionally, we have identified where in the overall architecture of a vehicle an indicator would fit. We have also proposed a centralized IDS with logic for the central node to combine the IOCs that one of them might not achieve the desired degree of confidence for raising an alarm. As part of the research, we have studied previous work in the field as well as interviewed industry experts. From this point, one could choose a subset of the IOCs for further evaluation and implementation.

Keywords: IDS, Intrusion, Detection, ECU, IOC

Acknowledgements

First of all, we would like to express our great thanks to our supervisors Tomas Olovsson from Chalmers University of Technology as well as Christian Sandberg and Manne Engelke from Volvo Group Trucks Technology. Without your generous contribution, we could not overcome the numerous obstacles we have been facing along the way to complete this project.

Second, we would like to thank the industry experts who have made invaluable contributions by willingly sharing their taxonomic expertise including Alex Ward, Andres Bokesand, Dhasarathy Parthasarathy, and Nasser Nowdehi. With your professional industry knowledge involved in the project, we got results of better quality.

Our sincere thank goes to all our friends who contributed with their valuable feedback and cooperation.

Last, but by no means least, we would like to thank our parents for their support and encouragement throughout our studies and lives.

Mohammad Hadi Sultani & Lu Han
Gothenburg, Sweden
June 10, 2019

Contents

List of Figures

List of Tables

List of Acronyms

ADAS	Advanced Driving Assistance Systems
CAN	Controller Area Network
CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Methodology for Information Technology Security Evaluation
CFI	Control Flow Integrity
D2B	Domestic Digital Bus
DoB	Denial-of-Body-control
DTC	Diagnostic Trouble Codes
EC	Evading Cost
ECU	Electronic Control Unit
FMS	Fleet Management Systems
GPS	Global Positioning System
HIDS	Host-based IDS
HU	Head Unit
IDE	Identifier Extended
IDPS	Intrusion Detection and/or Prevention System
IDS	Intrusion Detection System
IOC	Indicator of Compromise
IPS	Intrusion Prevention Systems
ITS	Intelligent Transport System
KWP2000	Keyword Protocol 2000

LIN	Local Interconnect Network
MAC	Media Access Control
MCU	Micro Controller Unit
MOST	Media Oriented Systems Transport
MPU	Micro Processor Unit
NIDS	Network-based IDS
OBD	On-Board Diagnostics
OEM	Original Equipment Manufacturer
OS	Operating System
PGN	Parameter Group Number
PKE	Passive Keyless Entry
RTR	Remote Transfer Request
SAE	Society of Automotive Engineers
SPN	Suspect Parameter Number
SSID	Service Set Identifier
SSL	Secure Sockets Layer
TCU	Telematic Control Unit
TLS	Transport Layer Security
TPMS	Tire Pressure Monitoring Sensor
TTCAN	Time Triggered Controller Area Network
TTP	Time Triggered Protocol
UDS	Unified Diagnostic Services
VAN	Vehicular Area Network
VLAN	Virtual LAN
VSS	Vehicle Speed Sensor
XCP	Universal Measurement and Calibration Protocol

1

Introduction

This chapter presents the context for the thesis project and justifies the importance of the subject by answering the question of "why to conduct this research?". Additionally, it defines the scope of the project to make it clear that "what is" and "is not" intended in this project. Finally, it presents an outline for the entire report and shows how it has been organized.

1.1 Context

Modern vehicles are no longer mere mechanical solutions. They are equipped with plenty of sensors and Electrical and Electronic (E/E) systems such as Electronic Control Units (ECUs), previously known as Engine Control Units. The E/E systems are used to control the primary functions in vehicles such as engine control, body control, transmission and braking systems as well as safety functions such as airbag, Advanced Driving Assistance Systems (ADAS), e.g., adaptive cruise control systems, and even entertainment systems. Additionally, some legislation mandates the use of ECUs in modern vehicles to make them comply with the safety and emission regulations, since the mechanical solutions such as legacy carburetors or hydraulic brake systems neither comply with such regulations, nor they can achieve the consumers' demands. Electronic Control Units are small computers with limited computational power but low energy consumption which are installed on board to control the systems better. Moreover, in order to achieve higher consumer satisfaction, vehicle manufacturers also implement advanced built-in entertainment or infotainment systems as well as navigation systems which in most cases require an Internet connection.

In order to communicate with each other, ECUs require to have a uniquely designed network to support the safety-critical functions. Such networks must be capable of performing in real time and must have bounded delays. Among many of such networks, Controller Area Network (CAN), proposed by Robert Bosch GmbH in the 1980s, has been widely accepted by vehicle manufactures due to its low cost of implementation and its bounded delay characteristic [2]. However, CAN was designed without having the potential cyber-security threats in mind.

Moreover, in order to program the ECUs, hundred of millions of lines of code are written [3]. The National Aeronautics and Space Administration (NASA) has performed a study on Flight Software Complexity which is developed by having the security as part of the design and is carefully tested, but still on average, two defects per 1000 lines of code remain in the software [1]. While most of the defects cause functionality glitches, some others can cause security issues. However, the number of defects is much higher in the applications which are not designed for safety-critical functions and are not tested as carefully as flight software. Vehicle ECUs have more unknown defects which can only be discovered if certain conditions are met. Figure 1.1 shows how defects are removed at different stages in flight software. A similar procedure is followed for programming the ECUs in vehicles as well.

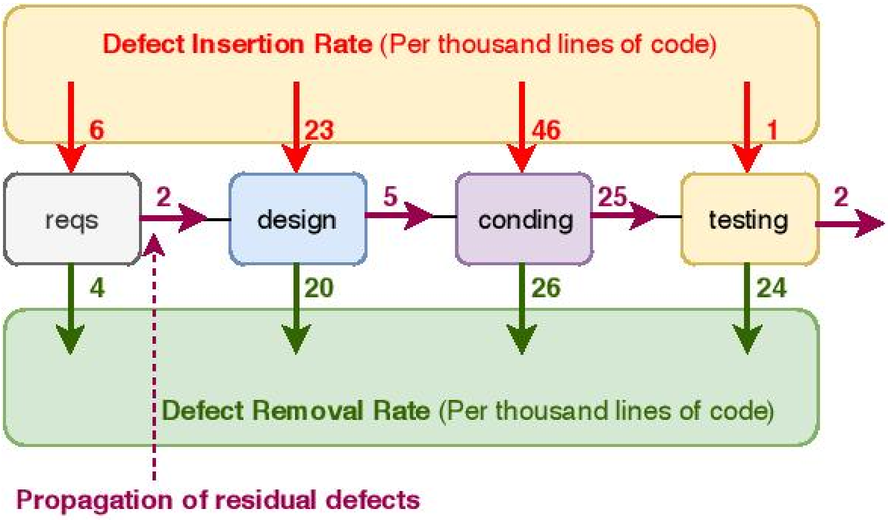
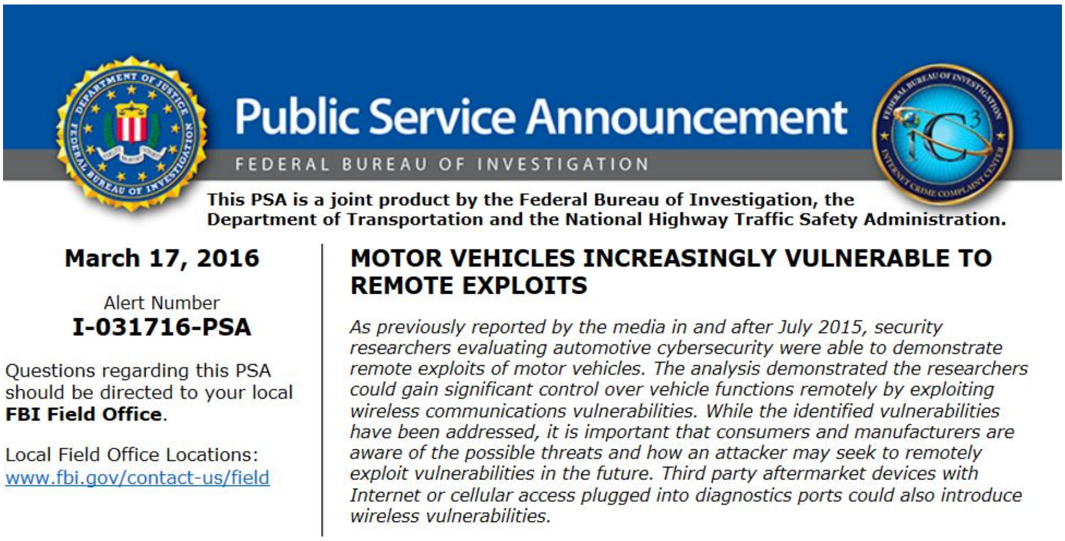

Defect Insertion Rate (Per thousand lines of code)
6
23
46
1
reqs
2
design
5
conding
25
testing
2
4
20
26
24
Defect Removal Rate (Per thousand lines of code)
Propagation of residual defects

Figure 1.1: Defects per 1000 Lines of Code [1]

Connection to the Internet, and having complex internal networks as well as complicated software have exposed modern vehicles to many cyber threats. Both intruders and researchers have compromised the security of vehicles in different ways either locally or wirelessly. Hacking wireless transmissions from Tire Pressure Monitoring Sensors (TPMS) and Key Fobs to taking control of the critical ECUs and modifying the ECUs' software are some examples of vehicles' security breaches [4]. Figure 1.2 shows a public announcement by the United States Federal Bureau of Investigation regarding the security of vehicles. This announcement was made in 2016 while very serious experimental vehicle hacks had been demonstrated back in 2010.

Significant security research on vehicular systems began in 2010, when a team of researchers, led by Prof. Stefan Savage from the University of California, San Diego,



Public Service Announcement

FEDERAL BUREAU OF INVESTIGATION

This PSA is a joint product by the Federal Bureau of Investigation, the Department of Transportation and the National Highway Traffic Safety Administration.

March 17, 2016

Alert Number

I-031716-PSA

Questions regarding this PSA should be directed to your local **FBI Field Office**.

Local Field Office Locations:
www.fbi.gov/contact-us/field

MOTOR VEHICLES INCREASINGLY VULNERABLE TO REMOTE EXPLOITS

As previously reported by the media in and after July 2015, security researchers evaluating automotive cybersecurity were able to demonstrate remote exploits of motor vehicles. The analysis demonstrated the researchers could gain significant control over vehicle functions remotely by exploiting wireless communications vulnerabilities. While the identified vulnerabilities have been addressed, it is important that consumers and manufacturers are aware of the possible threats and how an attacker may seek to remotely exploit vulnerabilities in the future. Third party aftermarket devices with Internet or cellular access plugged into diagnostics ports could also introduce wireless vulnerabilities.

Figure 1.2: US Government Public Announcement - Vehicle Vulnerability [5]

.

and Tadayoshi Kohno from the University of Washington experimentally evaluated the security issues of a modern vehicle and demonstrated that the underlying system structure is quite fragile [6]. Killing the engine and affecting the braking system were two of the most critical hacks since they involved the safety of the driver, the passengers, and other road users. However, their threat model required *prior physical access* to the vehicle, and this was viewed as unrealistic by others such as BBC [7] and Popular Science [8]. A year later, in 2011, in response to the criticisms made by others, the same team published another paper and systematically analyzed the *external* attack surfaces of a modern vehicle [9]. In 2015, Foster et al. [10] examined a Telematic Control Unit (TCU) which connects to the standard On-Board Diagnostics II (OBD-II) port. They demonstrated that such devices could be discovered, targeted and compromised by remote attackers. In fact, these devices are very popular and are used especially for Fleet Management Systems (FMS) as well as insurance firms to, among other things, be able to locate a vehicle and evaluate the driving behaviours of the drivers. J. Norte [11] has listed some vulnerabilities in such devices exposing vehicles to long-range attacks. Also, in 2015, Miller et al. [12] demonstrated hacking a 2014 Jeep Cherokee. Among other things, they managed to turn the steering wheel and activate the parking brake at highway speed. They continued their research and hacked the same model of Jeep in the following years to different extents. The results are discussed in [13][14][15]. These research papers made the security of modern vehicles a significant concern for vehicle manufacturers, vehicle security communities as well as the governments and proved that more research is required in order to mitigate such threats. Of course, there exist more hacking demonstrations than those listed above, but the fact has been proved that breaching vehicles' security could have severe consequences.

Although such threats exist against vehicles, many security researchers have already proposed different countermeasures for them. For example, Zhang et al. [16] proposed a Cloud-based anti-malware countermeasure. However, their countermeasure has raised up some privacy concerns, and the accuracy of the proposed countermeasure has not been measured. Wolf et al. [17] have proposed a cryptographic countermeasure to the issue. However, papers such as [18] has criticized the cryptographic methods, and they believe that such methods are not feasible because of the limitations that exist in CAN. A CAN message is only capable of carrying 8 bytes of data in a single frame, and the bandwidth is limited to 1 Mbps that is already reached in many practical scenarios. J1939 which is higher layer protocol based on CAN and is highly being used in heavy-duty vehicles, has only 25% of the bandwidth of CAN which has recently been upgraded to 50% which is only 500 Kbps. Refer to section 2.2 to read more on CAN, and section 2.2.3 to read more on the J1939 protocol. Some others have proposed Intrusion Detection System (IDS) as a countermeasure [19][20][21]. Intrusion Detection Systems for traditional computers and computer networks have been the focus of much research and their usage is widespread in IT industry. Nevertheless, IDS for vehicular networks has recently become the focus of many researchers but its applicability is still uncertain. In general, there are two alternative approaches that IDSs typically use to analyze sensors' data: Anomaly- and Signature-based detection [22]. Anomaly-based detection approach analyzes the current observed behavior of systems' use against the data relating to the behavior of legitimate purposes. This approach is able to detect zero-day attacks, but the number of false positives is still too high [22]. On the other hand, the signature-based approach applies a set of signature patterns of malicious data to the events in the system. This approach has minimized the number of false positives, but it is not able to detect zero-day attacks and requires significant effort to continually identify and review new malware to create their signature patterns and push them as update definitions to the end applications. Refer to section 2.3 to read more on IDS.

However, most of the research on IDS has been conducted for passenger cars while trucks have not been studied much. A significant difference between passenger cars and trucks is that truck configurations may be changed at any time even after the vehicle leaves the factory, e.g., at a workshop or a bodybuilder, resulting in a considerable amount of variants. As an example, a truck which leaves the factory might go to a bodybuilder that puts cranes and pumps on it or even rebuilds it into a fire or a refuse truck. This is considered a challenge in designing a security solution which works for all trucks.

1.2 Problem Statement

As discussed in the previous section, Anomaly-based IDSs are capable of detecting zero-day attacks but their most challenging problem is the number of false positives. Even with meager false positive rates, maybe one false positive per year, per vehicle, considering the number of vehicles deployed in the market e.g., one million, thousands of incidents need to be analyzed every day. A security operation center would be needed just to verify if there are intrusions or not. Additionally, several proposals for future legislation, in multiple markets, suggest the use of IDSs in vehicles to determine whether the vehicle is under attack and possibly take action based on that information [23][24]. Thus far, there are just proposals for legislation, and the Intrusion Detection parts seem more to be recommended rather than mandatory. However, in case the potential legislation mandates that an action must be taken based on the information provided by the IDS, then we must be confident that the IDS alarm is not a false positive.

In order to mitigate the number of false positives in the IDSs, we can improve them by using IOCs. Indicators of Compromise are simple artifacts or evidence whose presence in a system is a sign of intrusion or infection by malicious software. Regardless of the attacking technique, the IOCs trigger if a legitimate behavior of the system is violated. IOCs are also valuable because they can be used to prevent similar future attacks. An example of finding IOCs would be the use of honeypots in a vehicle, preferably a low interaction honeypot due to the limited computational resources on board. If it is contacted at some point in an intelligent way, the attacker will leave some indicators. Presence of such indicators is a reliable sign that the system has been compromised. An IOC could be reprogramming of an ECU if the vehicle is moving since such diagnostic routines would only be performed if the vehicle is in a maintenance workshop or at least if the vehicle is in a safe state.

1.3 Aim

In this thesis project, our aim is to find a reasonable subset of IOCs by observing the behavioral changes that attack would make in a vehicular system. We also map the IOCs to different layers in the overall architecture of a vehicle in order to determine the place that an IOC is expected to trigger. Additionally, we define a set of criteria and methodologies to evaluate the quality of the IOCs, and we categorize them into dependent and independent IOCs. Finally, we propose a methodology to combine several dependent IOCs to achieve the desired degree of confidence for raising an alarm if one of such IOCs cannot achieve the desired degree of confidence.

As the main goal of the research, we seek to answer the following research questions:

1. How to utilize IOCs to determine if an intrusion has taken place?
2. Where in the overall architecture will the IOCs fit?
3. How to evaluate the quality of IOCs?

1.4 Scope

In this thesis project, we are looking for the behavioral changes that an attack would make in a vehicular system, and define them as IOCs. While some IOCs in the list are able to indicate a system compromise independently, some others cannot confidently indicate a system compromise alone and must be combine with one or more other IOCs. We define a set of criteria and methodology to evaluate the quality of IOCs and categorize into two categories of dependent and independent IOCs. Since the focus of this thesis project is finding and evaluating the IOCs, so we do not implement any of them. We leave it as future work to further investigate the list of IOCs presented in this thesis project, and implement all or a subset of them in a vehicular system.

1.5 Methodology

To conduct this thesis project, we use the methodology described as follow:
To gain a deeper understanding of the concepts and technologies related to vehicular security, we perform a broad study of some literature, particularly the literature on Controller Area Network (CAN), its frame structure, its limitations from the cyber-security perspective, as well as the higher layer protocols based on CAN, e.g., SAE J1939.

To find the IOCs, we study many attack methodologies mostly performed by researchers which explain how they performed the attacks on vehicles, which strategies they used, and which vulnerabilities did they exploit. Furthermore, we observe which protocols features are missused, and what behavioral changes such attacks make in a vehicular system. Additionally, we study the literature on IDSs which attempted to address the security weaknesses in the vehicular systems. However, the primary purpose of studying the literature is not the countermeasures; instead, it is to find the vulnerabilities which the authors try to propose a countermeasure for.

In order to find the criteria and to qualitatively evaluate the quality of the IOCs, we have studied related literature such as Risk Assessment Frameworks and we have interviewed the industry experts in the vehicular cyber-security domain mainly at Volvo Group Trucks Technology, as well as some experts whom we met during the Vehicle Electronics & Connected Services (VECS) conference 2019. The primary goal of the interviews have been to include the industry professionals' perspectives in this academic piece of work.

1.6 Outline

The outline for the rest of this report is as follows: chapter 2 provides the technical background in which the necessary knowledge required to follow the concepts related to this thesis project are discussed. Chapter 3 presents a framework to describe different layers in the architecture of a vehicle and is used to map an IOC to one of those layers. Chapter 4 provides a list of the IOCs which we find by studying related previous works and interviewing the industry experts. Chapter 5 provides a set of criteria and methodology which are used to evaluated the quality of the IOCs. Chapter 6 presents the results of the thesis project. Chapter 7 discusses the results we get from Chapter 6, and finally chapter 8 concludes this thesis report.

2

Technical Background

The purpose of this chapter is to provide the reader with the insight and the technical background required to follow the concepts discussed in this thesis project. In section 2.2, we discuss the most widely used Automotive Internal Communication Network Technology, namely Controller Area Network – CAN. In section 2.2.3, we discuss the SAE J1939 which is a higher layer protocol based on the extended format of CAN developed by the Society of Automotive Engineers – SAE. Most of the heavy-duty vehicles including trucks, buses, and others use J1939 as a higher layer protocol to facilitate communication among the ECUs. In section 2.3, we discuss the Intrusion Detection Systems and their related concepts along with their current issues. Finally, in section 2.5, we provide a discussion of previous related works to give the reader the insight into the current state of the art in the vehicular security domain research.

2.1 Automotive Network Technologies

There are some characteristics which mandate the use of specialized protocols for in-vehicle network systems (bus) instead of the conventional computer network technologies such as Ethernet or TCP/IP. However, there exists a specialized version of Ethernet designed to be used for vehicular systems. The shortest time of message delivery, a guarantee of message delivery, minimum cost, non-conflicting messages, and resilience to the electromagnetic field are some of the required characteristics. Many of such specialized protocols exist such as Local Interconnect Network (LIN), CAN, Media Oriented Systems Transport (MOST), FlexRay, Bluetooth and a few more. However, each of the bus systems as mentioned above is designed for a special purpose. While LIN, which is a single-wire, single-master bus system, is mostly used as a sub-bus in vehicles, CAN is a two-wire and event-triggered bus used for the soft real-time system. FlexRay is a time-triggered bus with higher bandwidth than CAN and is used in hard real-time systems. MOST is used for multimedia functions. In fact, each of the aforementioned bus systems is a representative for a function group of vehicular network technologies. For each function group, there exist other protocols as well such as VAN (Vehicle Area Network), TTP (Time-Triggered Protocol), TTCAN (Time-Triggered), D2B (Domestic Digital Bus), and others, but the discussion of those protocols is beyond the scope of this thesis project.

However, CAN is still the dominating protocol, especially in the vehicular industry domain. Therefore, for this thesis project, we focus more on CAN. CAN FD (Flexible Data-rate) and TTCAN are the more updated versions of CAN and are used where the standard CAN seems to provide insufficient bandwidth and services.

2.2 Controller Area Network

Controller Area Network (CAN) was officially introduced by Robert Bosch GmbH in 1986 as a serial communication protocol mainly for automotive industry [2]. However, CAN also found its way in other applications where microprocessors need to communicate with each other. The specification of Bosch GmbH for CAN introduces CAN as “Controller Area Network (CAN) is a serial communication protocol which efficiently supports distributed real-time control with a very high level of security.” When it comes to reliability, the **probability of undetected error** in CAN is 1 in 1000 years [25]. Nevertheless, CAN provides no security from the cyber-security point of view since it has been designed for communication between "trusted parties" only. By design, CAN does not even provide the basic principles of security, the so-called CIA triad: Confidentiality, Integrity, and Availability. Due to the lack of CIA as part of the design of CAN, it is not possible to identify the source ECU or the message generator. Additionally, it is not possible to detect modification of message content, and it is easily possible to perform Denial of Service (DoS) attacks on the network. Moreover, because all messages sent by a node are broadcast in the network, compromising one vulnerable node can potentially jeopardize the network as a whole. Researchers have shown that by accessing a single CAN node, they could successfully inject messages into the network and perform actions which are not normally allowed [13].

CAN supports four types of frames:

- Data frame
- Remote frame
- Error frame
- Overload frame

In the rest of this section, we dive deeper into the details of the Data and Remote frames, but we do not cover the details of Error and Overload frames as they are beyond the scope of this thesis project. In fact, Data and Remote frames are mostly the same except the RTR-bit (Remote Transfer Request) which in a Remote frame is set to a recessive (high) value and a Remote frame has no Data field [25].

In 1995, the extended format of CAN was also published as an ISO 11898 amendment [25]. The standard format of CAN (2.0A) supports an 11-bit identifier. A message identifier (ID) is used for identifying the type of messages as well as for prioritizing the messages transmitted over the network so that time-critical messages can meet their timing deadlines. However, the extended format of CAN (2.0B) supports a 29-bit ID to support more message types. While the standard format can only support up to $2^{11} = 2048$ different message types, the extended format is able to support up to 2^{29} which is more 536 million types of messages. Both formats can exist over the same bus. The IDE-bit (Identifier Extended) is used to differentiate between the standard and the extended formats. If the value of this bit is a dominant (low), then the frame is of standard format, but if it has a recessive (high) value, the frame is of extended format.

2.2.1 Message Frame Format

A Data frame consists of seven-bit fields. Start of Frame (SOF), Arbitration, Control, Data, Cyclic Redundancy Check (CRC), Acknowledgement (Ack), and End of Frame (EOF) fields. Interframe Space consists of three bits which are, technically not considered as a bit field of CAN frame but, transmitted as recessive bits to indicate that the bus is idle. The remote frame is used by an ECU to request another ECU to send some data to it. Table 2.1 contains the details of a standard CAN Data frame.

Field Name	**Size (bits)**	**Comments**
Start of Frame	1	Must be a dominant bit
Arbitration	12	11 bits Identifier ID10 - ID0. 1 bit RTR
Control	6	1 bit IDE, 1 bit reserved r0. 4 bits Data Length Code DLC3 - DLC0
Data	64	From 1 to 8 Bytes
CRC	16	15 bits CRC Suquence CRC14 - CRC0. 1 bit Delimiter, must be recessive
Ack	2	1 bit ack Slot. 1 bit Delimiter, must be recessive
End of Frame	7	7 bits, all must be recessive

Table 2.1: CAN Data Frame (Standard)

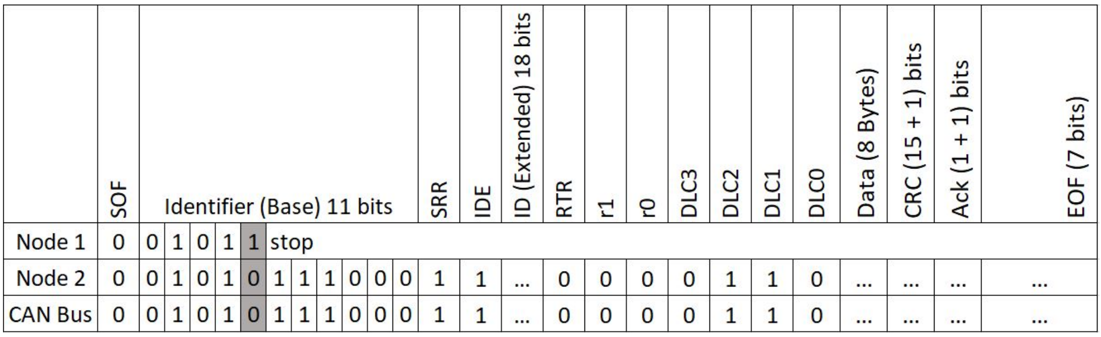
Table 2.2 shows only the Arbitration and Control fields of an extended CAN Data frame. The remaining fields are identical to the standard format. For more detailed information, refer to [2] and [25].

2.2.2 Arbitration

When the CAN bus is idle, any node is allowed to start transmitting its message over the bus. However, in the case of two nodes starting to send messages simultaneously, the message prioritization (or Arbitration) takes place that decides which of the nodes can continue sending its message and which node must immediately stop. The arbitration is important for time-critical messages to be delivered before their deadlines expire. Figure 2.1 shows the process of arbitration when Node 1 and Node 2 are trying to transmit their messages at the same time. Node 1 transmits its message with ID = 0x2FA and Node 2 transmits its message with ID = 0x2BC. As the figure shows, at bit 5th of the Identifier, Node 2 wins the arbitration and is allowed to proceed. This is because a dominant (0) bit always wins over a recessive (1) bit. Node 1 is stopped at this point and must wait until the bus is idle again. This figure is also a compliment to the content of Table 2.1 and Table 2.2 by showing the bit fields of a CAN frame. Figure 2.1 is also a compliment to the tables 2.1 and 2.2.

Field Name	**Size (bits)**	**Comments**
...	...	...
Arbitration	32	11 bits Identifier (base) ID28 - ID18. 1 bit SRR, 1 bit IDE, 18 bits Identifier (extended) ID17 - ID0. 1 bit RTR
Control	6	2 bits reserved for future r1 - r0. 4 bits Data Length Code DLC3 - DLC0
...	...	...

Table 2.2: CAN Data Frame (Extended)



	SOF	Identifier (Base) 11 bits											SRR	IDE	ID (Extended) 18 bits	RTR	r1	r0	DLC3	DLC2	DLC1	DLC0	Data (8 Bytes)	CRC (15 + 1) bits	Ack (1 + 1) bits	EOF (7 bits)
Node 1	0	0	1	0	1	1	1	stop																		
Node 2	0	0	1	0	1	0	0	1	1	1	0	0	0	1	1	...	0	0	0	0	1	1	0	...	...	...	...
CAN Bus	0	0	1	0	1	0	0	1	1	1	0	0	0	1	1	...	0	0	0	0	1	1	0	...	...	...	...

Figure 2.1: Arbitration Process. (Extended CAN Frame)

2.2.3 SAE J1939

SAE J1939 is a higher layer protocol which is designed based on CAN. Similar to CAN, it provides serial communication among the ECUs in almost all kinds of heavy-duty vehicles. Nonetheless, it only supports the 29-bit identifier CAN. Extending the CAN identifier was the result of a request by the SAE to support the J1939 standard. While CAN supports up to 1 Mbps of data rate, J1939 only supports up to 250 Kbps. However, in October 2011, SAE published the J1939/14 standard which states that it can support up to 500 Kbps [26]. Additionally, J1939 only supports up to 30 ECUs and a maximum of 253 Controller Applications – CA [27]. One ECU is capable of managing several CAs. In contrast to CAN which is not concerned regarding the source and destination addresses, J1939 utilizes the 29-bit identifier to identify the source and the destination addresses of ECUs. In fact, it is mandatory for ECUs to hold one NAME and one ADDRESS for identification purposes. This is a crucial feature which assists in identifying the potential attacks sourced from a compromised ECU which is usually not allowed to communicate with other specific ECUs. Also, J1939 specifies how to handle multi-packet messages with a maximum size of 1875 Bytes when data larger than 8 Bytes needs to be transferred.

J1939 also specifies how to convert raw data into human-readable data. J1939/71 is a document with roughly 800 pages with the definitions of Parameter Group Numbers and Suspect Parameter Numbers. J1939 has introduced the concept of Parameter Group which is a group of related data. For example, Engine Temperature is a group of related data such as Engine Coolant Temperature, Fuel Temperature, Engine Oil Temperature, etc. Each of these groups is assigned a number which is called Parameter Group Number (PGN) and the data themselves which are transferred into the Data field are referred to as Suspect Parameter Numbers – SPN. For simplicity, Parameter Group Numbers and Suspect Parameter Numbers can be called Messages and Signals respectively. PGNs are helpful in identifying message types. This is crucial, for example, we can see diagnostic messages on the network while a vehicle is running.

2.3 Intrusion Detection Systems

To protect the computers and networks against intrusions, IDSs are used which only detect the intrusions and raise an alarm in case an intrusion is detected. Intrusion Prevention Systems (IPS) take the IDS to yet another level and take action if any intrusion is detected. The two terms together are called Intrusion Detection System and Intrusion Prevention System (IDPS).

In general, there are two alternative approaches that IDSs typically use to analyze sensors' data: Anomaly- and Signature-based detection [22]. Anomaly-based detection approach analyzes the current observed behavior of systems' use against the data relating to the behavior of legitimate uses. This approach is able to detect zero-day attacks, but the number of false alarms is still too high. A false alarm can be either a False Negative or a False Positive. The former happens when an intrusion is missed by the IDS so it does not raise an alarm (negative) while it had to do so (false). The latter is when legitimate traffic flow is tagged as an intrusion (positive) while it should have not been the case (false). Due to the fact that False Positive alarms require a manual evaluation, a high number of such alarms will be time-consuming, resource demanding and adds to operational costs. This is one of the main reasons that False Positive alarms is the main focus of security research. False Negative can be considered as the sensitivity of a system and can be minimized after a substantial decrease of False Positive alarms by obtaining a better understanding of the legitimate behavior of the system. On the other hand, the signature-based approach applies a set of signature patterns of malicious data to the events in the system. This approach minimizes number of false positives, but it is not able to detect zero-day attacks and requires significant effort to continually identify and review new malware to create signature patterns and push them as update definitions to the end applications.

2.3.1 Placement of IDS

In addition to the type of detection, IDSs can be categorized based on the level in which they operate. A Host-based IDS (HIDS) is installed on the sensitive or vulnerable hosts; examples in Information Technology (IT) industry could be a database server or other administrative systems, and in the vehicular industry it could be a sensitive ECU. A Host-based IDS can add an extra layer of security to the system. In the case of IPS, an intrusion or attack (both external and internal) could be stopped on the spot or at least a log entry would be created for the incidence. A Distributed HIDS is another technique which, instead of mounting stand-alone HIDS on every single host, makes coordination and cooperation among them in the network.

Similar to a Host-based IDS which only monitors the activities within a single host, a Network-based IDS (NIDS) monitors the data traffic at selected points and raises an alarm if an intrusion is detected. Depending on the level of inspection implemented in a NIDS, it examines different levels such as Network, Transport, and/or Application.

2.3.2 Limitations

There are some problems with IDSs which make them not so practical in the vehicular domain. Among them, False positive alarms is one the most serious ones due to the massive number of vehicles in the market. Adaption/portability is the second problem. Since the users' behavior varies, it is not possible to develop an IDS which can be adapted to all systems. The services are often unique, and adaptation of IDSs requires some time. The third problem is the scalability of the IDS. Network speed plays a significant role in this matter because the number of sensors and analyzers will be different for different network sizes. Privacy concerns have also emerged as an obstacle and a problem for the IDSs. Where the files will be scanned for intrusions and what other processes will be performed on the users' private files have become a challenging concern.

2.4 Common Criteria

The Common Criteria for Information Technology Security Evaluation (CC) is an international program, which is also maintained as an ISO/IEC 15408 standard [28], and is used as a framework to certify IT products for computer security measures. This framework contains Security Functional Requirements (SFR) which consumers, developers, and evaluators can use to assure if a claimed security level for an IT product is fulfilled. It also contains Security Assurance Requirements (SAR) which shows the depth of the evaluation which has been carried out in an evaluation laboratory. The evaluation is usually carried out by a private entity and is supervised by a governmental organization. Since CC is only a large set of criteria, it is accompanied by the Common Methodology for Information Technology Security Evaluation (CEM). CEM is maintained as an ISO/IEC 18045 standard [29] and it contains the methods by which the evaluation should be carried on considering the evaluation criteria from CC. We have used CEM to evaluate the feasibility of the attacks which lead to the IOCs provided in this thesis project.

It is worth mentioning that the ISO/IEC 15408:2009 and ISO/IEC 18045:2008 are the latest published versions of the standards, but the most recent version (2017) of CC and CEM can be obtained from the official website of Common Criteria.

2.5 Related Work

Indicators of Compromise in the context of the IT industry has widely been studied. However, to the best of our knowledge, IOCs have not been the focus of much research in the vehicular domain. Catakoglu et al. [30] have proposed a methodology to extract the IOCs for Web Applications automatically using high interaction honeypots installed on some virtual servers. Although this work is not fully relevant to our research, we still could use some of the concepts which we believe are common in both domains. In the rest of this section, we provide some related works which propose countermeasures for the cyber-security issues in the vehicular domain.

Wolf et al. [17] have proposed a countermeasure for the security issues in automotive bus systems based on modern cryptographic mechanisms which, in theory, resolves the secrecy, manipulation and authentication issues. However, the proposed countermeasure adds some extra overhead and requires more computational power but does not take into account the bandwidth limitation in some network technologies such as CAN which is the de-facto standard for in-vehicle bus systems. Müter et al. [31] proposed an entropy-based anomaly detection approach for the IDSs to be used for vehicular bus systems which seems to have high accuracy for the test cases they evaluated. However, there still the possibility that the mentioned approach raises a false positive alarm which puts the applicability of the approach under question.

3

Framework

In order to facilitate communication among dozens of ECUs which are typical in the modern vehicles of today, different network technologies are used by the manufacturers. While CAN is a dominating network technology for the components which require real-time communication, other network technologies are also being used such as LIN, MOST, and others which we have explained in chapter 2. Each one of the aforementioned network technologies requires different security mechanisms. Thus, different levels of security must be implemented at different layers to secure a vehicle against compromise. NXP Semiconductors N.V. (NXP, for short) which is a Dutch global semiconductor manufacturer, proposes a 4 + 1 security framework. The 4 layers are Secure Interfaces, Secure Gateway, Secure Network, Secure Processing, and the +1 is securing the traditional physical access to the vehicle. The +1 layer is not of use for our thesis project, but we found the 4 layers part useful. We use this framework not for the purpose that it is designed, but as a general framework that we can divide an automotive system to Interfaces, Gateway, Network, and Processing layers. These are the four layers in which we can potentially see an IOC. We explain and discuss on IOCs in chapter 4.

3.1 Layer 1 - Interfaces

Researchers [12][13][14][15] [6][9] have shown that the communication network inside a vehicle is not secured in most of the cases, and the communication with the outside world is also exposed to potential attacks. The vehicles' external wireless interfaces present attack surfaces for hackers. If an attacker explores the interface and gets access to the internal network somehow, it is possible to perform any malicious activities, such as downloading a user's private information or even control the vehicle. Therefore, the first layer of protection should be securing the interfaces. However, for the sake of this thesis project, we are only concerned about finding the IOCs which can be a result of exploiting the interfaces. Figure 3.1 shows a simplified diagram of the typical components of a vehicle. Telematics Control Unit (TCU) and On-Board Diagnostic are considered as Interfaces in the diagram that we could potentially expect an attack from. Should we find any IOC that can be a result of exploiting this layer, we will mention it in chapter 4. Although OBD-II port can be physically secured, there are tons of third-party devices which connect to this port and provides wireless connectivity to the vehicle. Lots of companies use such devices as part of their Fleet Management System, for example, to monitor the driving behavior of their drivers and fuel consumption levels.

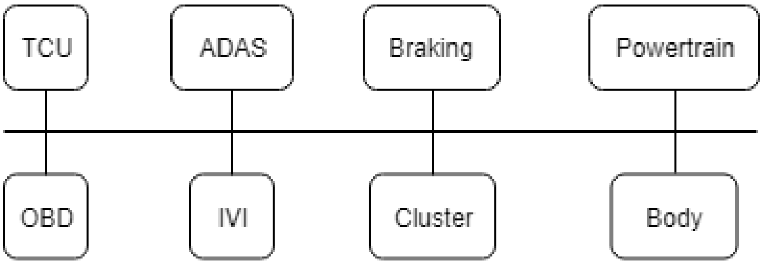


Figure 3.1: Layer 1: Interfaces

3.2 Layer 2 - Gateway

When Jeep was hacked in 2015 for the first time [12], it turned out that if a hacker would explore the interfaces, it was possible to reach to any destination without limitation, for instance, the ADAS system. A centralized gateway divides the network into the interfaces of the outside world and the inside main function domains. It acts as a firewall that controls the communications between the external interfaces and the internal network by allowing or denying the ongoing traffic. A gateway provides physical domain isolation between the untrusted infotainment systems and trusted safety-critical systems.

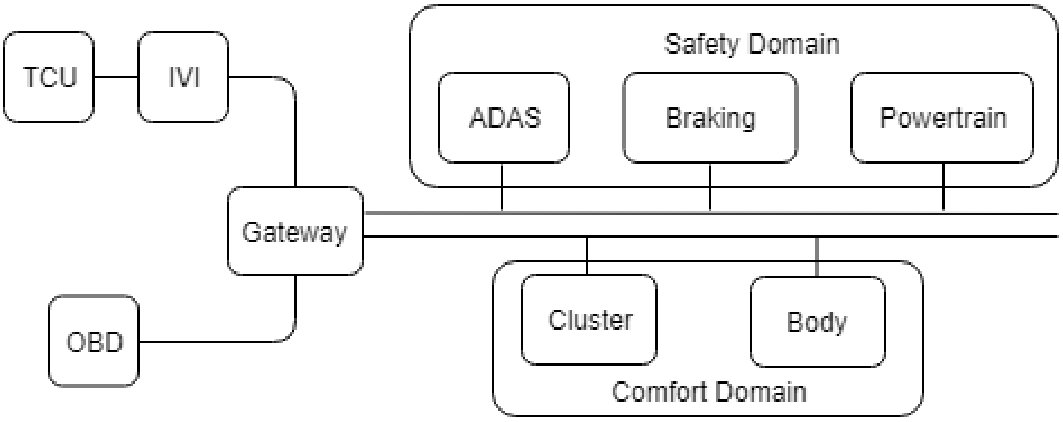


Figure 3.2: Layer 2: Gateway

3.3 Layer 3 - Network

For security reasons, most of the manufacturers use subnetting techniques to isolate the safety-critical sections of the network from other parts. Should an attacker get access to a subnet, it is usually difficult to communicate with other subnets due to the existence of a firewall between the subnets which checks packet integrity. Determining that an attack is happening at a particular subnet, can be useful. However, in some cases, a single IOC might not explicitly tell us which ECU is compromised but knowing which subnet is under attack can help us isolate the attack. Additionally, the applied security level of the subnets may differ due to the functionality difference. Although those subnets might be of more interest for an attacker, the unsecured subnets are attacked because of the potential ease of work. Figure 3.2 shows two subnets of a vehicular network, namely "Safety domain" and "Comfort domain."

3.4 Layer 4 - Processing

The data transmission between different transceivers is secured in layer 3, another layer of security can be securing the software installed on the Micro Controller Unit (MCU) and Micro Processor Unit (MPU). As section 1.1 explains, modern vehicles have hundreds of millions of lines of code which, almost, always have bugs and security flaws. With such complex software systems, the number of defects is also high. When some of the defects are found after the vehicle leaves the factory, the Original Equipment Manufacturer (OEM) should have remote access to the software and perform Over The Air (OTA) upgrades in order to patch the issues found. Also, by implementing secure boot and run-time integrity to microcontrollers guarantees the code is authentic. However, that is not always the case, the security bugs in the code are exploited to attack ECUs. We will discuss some IOCs related to layers in chapter 4.

3.5 Secure Access

Before the Jeep was hacked in 2015, Charlie Miller and Chris Valasek actually performed a similar attack with physical access to Jeep in 2013 [12], based on that experience, more and more complex remote hacking was tested. Even though Stephen Checkoway et al. said, it’s not realistic that an attacker has physical access to the vehicle, but if the physical access is available, non-computerized attacks can be performed e.g. cutting the brake line [9], the physical access is still critical for protecting a vehicle. In addition to physical access, some other accesses like remote lock/unlock, remote vehicle monitoring using smartphone or wearable devices should also be considered when securing access to a vehicle system.

3.6 Secured HoliSec Reference Architecture

A vehicle reference architecture is introduced by Atul Yadav and Christian Sandberg in the HoliSec project in 2018 [32, 33]. By combining the in-vehicle architecture from the HoliSec project with layer 1 to 4 secure mechanisms, a secured holistic architecture is presented (Figure 3.3). The new architecture not only considers the potential security issues, but also tries to increase the performance. First of all, the framework combines several network protocols. Low bandwidth communication technology CAN provides secure communication between nodes in the network, while the Ethernet provides a greater bandwidth. Second, this architecture uses Virtual LAN (VLAN) to divide the physical layer into virtual networks to guarantee the transmission of critical data. Finally, this secured HoliSec reference architecture secures the interfaces, gateway, networks and data processing according to the NXP secure schema. So this new architecture is used as a framework in this thesis work.

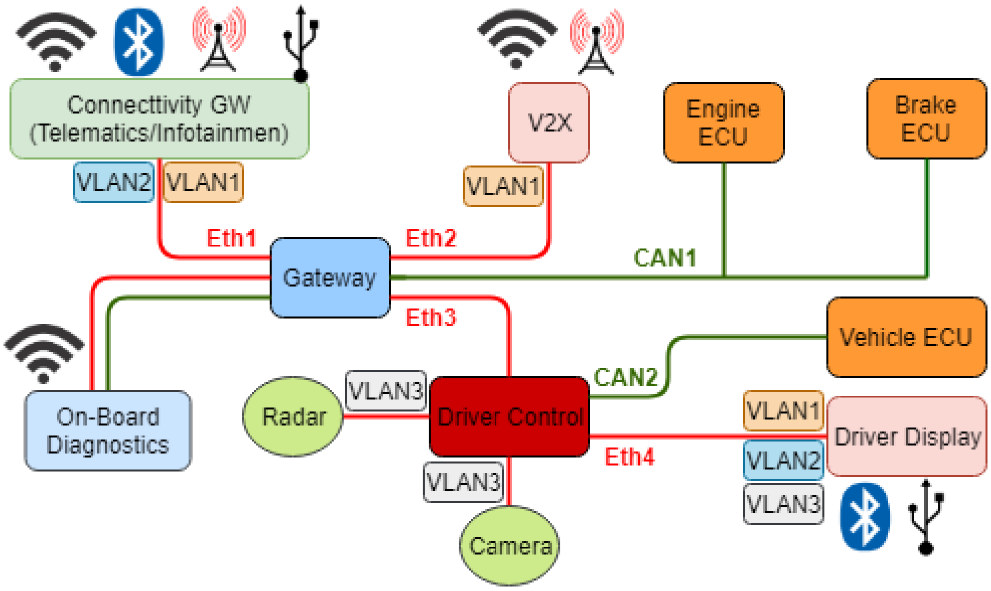


Figure 3.3: HoliSec Reference Architecture

4

Indicators of Compromise

IOCs in the traditional IT industry has been the focus of much research, but this is a relatively new concept in the vehicular industry. In this thesis project, we have tried to transform any IOC concept from the IT industry into the vehicular domain when applicable.

In order to find a list of IOCs, we studied lots of attacks mostly performed by researchers on vehicular networks. Then, we have used the behavioral changes which a successful attack would make in a vehicular system as an IOC. For example, injection of a message into the network is a type of attack, which as a result, the frame frequency of the injected message increases. So, changes in the frame frequency are considered to be an indicator of message injection, which in turn, is an IOC. However, some indicators can explicitly tell which part of the system in being compromised, some others need to be combined with one or more IOCs in order to be useful. Additionally, we have interviewed some industry experts in order to have a professional industrial perspective into this thesis project as well.

In this chapter, we provide a list of IOCs for the vehicular systems. The IOCs presented in this chapter are divided into four categories according to the layers introduced in chapter 3. Furthermore, the IOCs in the Network layer can be a result of exploiting at least one of the eight factors that we discuss later in this chapter.

4.1 Interface Layer IOCs

The Interface layer encompasses all connectivity interfaces that provide a means to connect the vehicle to an external device or to the world via Internet. The threat models that require a physical access have met significant and justifiable criticism that a presuppose of the attackers' access to the internal components of a vehicle is unrealistic. The HoliSec Reference Architecture (figure 3.3) has modeled four different connectivity means for a vehicle, USB (physical access required), Bluetooth (short range), as well as WiFi and Cellular (long range). It is worth mentioning that the mentioned four connectivity means are typical in almost all vehicles of today. In this thesis project, our focus is on the long-range connectivity interfaces which are WiFi and Cellular. Having a secure Interface layer is crucial in having a secure vehicular system as this is the entry point for all remote attacks.

In this section, we provide a list of IOCs that we expect to see in this layer of a vehicular system.

4.1.1 Port Scanning of the Connectivity Gateway

A TCP network port is a 16-bit number making a total of 65,536 (0 - 65,535) possible port numbers. In the traditional computers, the number of open ports used by different applications and services vary a lot and can also change from time to time since new applications or services that need to communicate over the network can be installed or removed by the user at any time. However, the number of ports that must be open in vehicles is limited to only a few ports, e.g., 5 out of 2^{16} ports since a vehicle owner cannot install new services or remove the existing ones as easily as in a traditional computer. It is worth mentioning that an open port is only usable if a service or an application is actively listening on that port, which is what an attacker is interested in. Additionally, attackers are very interested in port numbers below 1024 (0 - 1023) that are called system or well-known ports [34] because they are often mapped to well-known services. In order to perform the port scanning, there exist several different methods but a discussion of such methods is beyond the scope of this thesis project. Furthermore, an attacker will most probably not scan all possible port numbers because of the time required.

From an IOC prospective, port scanning is a strong IOC that if seen, the likelihood of an attack is very high since a legitimate user who needs to connect to the vehicle, for any reason, already knows the open port number and does not need to scan them; thus only someone with a malicious intention might perform a port scan. Therefore, we do not expect to see a port scanning activity at all. Also, the time required to perform a port scanning is almost static. We will use this fact as part of the evaluation of IOCs in chapter 5. While performing port scanning (all possible port numbers) using OpenVAS (Open Vulnerability Assessment System) on a traditional computer running Linux Operating System (OS), it took almost one hour time for us. Of course, lots of factors can affect this time such as the processing power of the scanner machine and the network speed. If implemented, this IOC can also help in recognizing real-world attacks against vehicles because almost all attacks performed against vehicles today are research based.

4.1.2 Evil Twin SSID Existence

Vehicle manufacturers connect their TCUs to the HUs in two different ways, wired or wireless. The reason for the existence of such a connection is to transmit the information from a TCU to the HU, which is installed on the dashboard of a vehicle. Such information can be navigation or diagnostic data as well as Internet connectivity. These days, most TCUs are capable of providing WiFi hotspots to driver and other passengers by having a SIM card and a connection to a cell phone base station. Additionally, not only the TCUs are connected to the HUs wirelessly, but also in the trucks with long chassis, to reduce the cost of wires and to decrease the complexity of the systems, many sensors connect to a central point wirelessly.

An attack against the TCUs (and any other sensors that connect to another part of the vehicle wirelessly) that connect to the HU through WiFi is to fool it to connect to an SSID with the same name as the HU that belongs to a different device. In a normal scenario, the TCU connects to the HU where the SSID is usually hidden (for security reasons) and its signal strength is set such that the broadcasting range is short. First of all, an attacker needs to find the hidden SSID, which can be done by listening to the TCU that broadcasts the SSID when trying to connect to the HU. Then, the attacker creates the same SSID (evil twin) with a stronger signal to make the TCU to connect to the evil twin. At this point, the attacker has a connection to the TCU and has the opportunity to continue to the attack. Although manufacturers have different naming schemes for the SSIDs, they try to make the SSIDs unique to a vehicle e.g., by using the chassis number as the SSID and prepending or appending some string to make it unguessable. Therefore, we consider the existence of the same SSID at the same place as an IOC.

4.1.3 TCU New MAC Address Connection

Related to the above IOC, is when the TCU connects to the evil twin SSID, the MAC address will be different than the expected MAC address of the HU which can be hard coded in the TCU. This could be a very strong IOC, but due to the feasibility of MAC address spoofing by the attacker, the strength of this IOC is reduced.

4.1.4 Link Downgrade

Nowadays, TCUs support SIM cards and are capable of connecting to cellular base stations similar to an ordinary cell phone. One attack against such TCUs is that the attacker creates a fake base station, so that the TCU connects to that instead of the real base station. At this point, the attacker can act as a man-in-the-middle. It is worth mentioning that this attack is against all cellular communications and researchers are actively working on the creation of countermeasures. For instance, Ericsson has an article on how to detect the fake base stations [35]. Taking this attack one step further is to jam the signals of 3G and 4G to force the TCU to switch to 2G that does not have any encryption mechanism. This way, the attacker can easily read all the data being transmited by the TCU. While some researchers believe that 2G should no longer be supported, some others consider it necessary for emergency situations.

4.2 Gateway Layer IOCs

In the new vehicular network topology, Gateways are used to make the complex networks more manageable as well as to enhance their protection level against cybersecurity attacks. Division of such networks into subnets is mostly performed based on the nodes functions. For example, safety-critical components such as Engine and Brake ECUs can be part of a safety-critical subnet. Additionally, a vehicular network is comprised of different network protocols as discussed in chapter 2. Therefore, Gateways in a vehicular network usually perform two primary functions, namely frame filtering and protocol translation. Lack of Gateways in the architecture of the 2014 Jeep Cherokee made the life of hackers (Dr. Miler and Valasek) much easier as they state in their paper [12]: "... there are no CAN bus architectural restrictions, such as the steering being on a physically separate bus. If we can send messages from the head unit, we should be able to send them to every ECU on the CAN bus."

In this section, we provide a list of IOCs that we expect to see in this layer of a vehicular system.

4.2.1 Information Presence on an Unexpected Network

When a gateway is involved in the architecture of a vehicle and the network is broken down into subnets, then not all ECUs are supposed to communicate with each other. For instance, an infotainment system should not communicate with the safety critical ECUs such as engine or brake ECUs. ECU domains are separated, so we don't expect to see unexpected frames in different networks. Existence of such unexpected frames is considered an IOC.

4.2.2 Integrity Check Failure in Routing Rules

As discussed above, a vehicle network is broken down into subnets for the sake of security and performance. However, some ECUs from different subnets are still required to communicate with each other. For example, a TCU might read some diagnostic messages from the safety-critical ECUs in order to perform a state-of-the-health checks. In such cases, the routing rules allow them to communicate, but a failure in the integrity check of such rules can be indicator of compromise since the legitimate ECUs which are not supposed to communicate beyond their authorization, would not do so.

4.3 Network Layer IOCs

A compromised network can endanger all connected nodes. In order to provide a comprehensive list of IOCs for the Network layer of an automotive architecture, we have used the eight factors determined by Müter et al. in their anomaly-based IDS paper [36]. While checking for anomalies, they define eight sensors that each checks for the Formality, Location, Range, Frequency, Correlation, Protocol, Plausibility, and Consistency of message(s). We have used each of the mentioned sensors as an attack type, and each of the IOCs described below, belongs to at least one of the eight sensors. Also, we have listed at least one IOC for each of the eight factors making our list of IOCs a comprehensive list that covers all potential types of attacks on the Network layer. At the end of each subsection below, we mention one or more of the eight factors mentioned above, to which the IOC relates. However two of the factors, namely Location and Correlation are already covered under Gateway layer IOCs so they are not covered here.

4.3.1 Unexpected Physical Characteristics of Signals

Depending on the protocol being used for the network communication in a vehicular system, characteristics of the signal might differ, but they are always defined in a formulated manner. For example, in case of CAN, a remote frame must have its RTR bit set to a recessive value and the payload must be empty. If a frame which has the RTR bit set to a recessive value and also contains a payload, then the physical characteristic or the formality of the frame has been violated. Such violations are considered an IOC.

Factor: Formality

4.3.2 Out-of-Range Signal Values

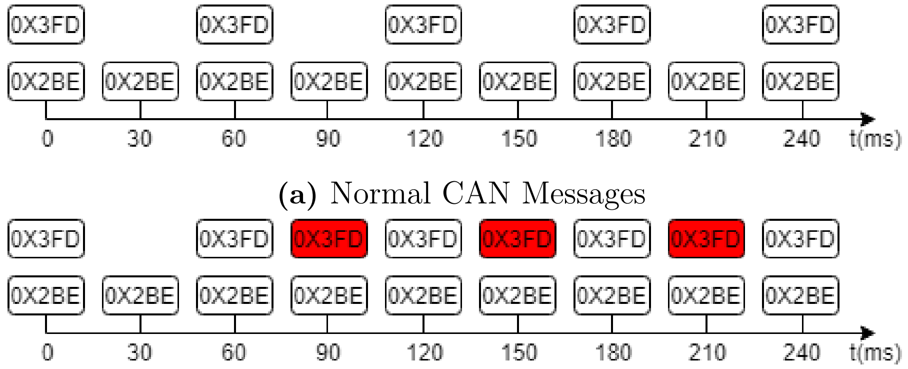
This IOC looks into the payload of the messages being transmitted on the network. Although, the Data field in CAN protocol allows 8 bytes of data to be sent in each frame, in reality not all combinations are used. For example, if the maximum possible speed of vehicle is 200 km/h and a one-byte data field is used which can show a maximum speed of 255 km/h $2^8 = 256$ $(0 - 255)$, any value above 200 is out of the possible or expected range, hence can be an IOC.

Factor: Range

4.3.3 Frame Frequency Increase

Most of the ECUs connected to the CAN network sends messages periodically. However, some of them have a range within which the frames are expected to be generated. Some other messages do not have any period, and are sent only when needed e.g., pressing the button to activate the cruise control function. Since most of the messages are sent periodically, injecting new messages causes a frame frequency change. Injecting a single CAN ID, injecting pre-ordered messages of multiple CAN IDs, and injecting a massive number of CAN messages are the three types of message injection which Song et al. define in their paper [20]. Each type of message injection has its own specific purpose, but the common result is frame frequency change on the network. The important point to consider is that a frequency change is not specific to any vehicular network technology e.g., CAN or Ethernet and it is possible to monitor the frequency on all of such network technologies. Figure 4.1 shows how a message can be injected into the network. Figure 4.1a shows that the message 0x2BE is being generated every 30 ms while the message 0x3FD is being generated every 60 ms by their respective ECUs. Figure 4.1b shows that after the second message ID of 0x3FD, a new message is being injected every 30 ms – red messages. This injection will double the frequency of the message ID 0x3FD since the injection period is half of the legitimate period.

Several different methodologies exist to counteract this type of attack. Song et al.



(a) Normal CAN Messages

(b) Injected CAN Messages

Figure 4.1: Message Injection

have tried the Time Interval Analysis method where they calculate the arrival time of a message and compare it with the latest arrival time of that message. If the interval of the message is shorter than half of the normal interval, the message is considered to be injected. However, there exist smarter ways of message injection in which the time interval might not be half of the normal interval. Instead, the injection might very slightly change the frequency, making it very hard to be detected. In such cases, detection of the injected message would be much harder. Furthermore, other researchers such as Müter et al. taken another approach where they calculate an entropy value for the network first [31]. After the message injection, they recheck the entropy, and if the new value is less than the initially calculated value, then it means a message injection. However, to tell precisely which message identifier has been injected, they use the concept of relative entropy and calculate the relative distance between two data sets which are defined over the same ID. Both of these methods are able to detect the message injection shown in figure 4.1.

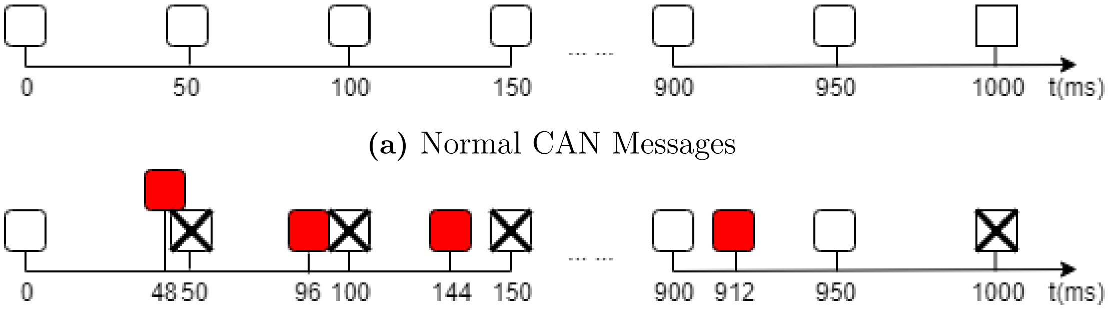
From an IOC point of view, although a frame frequency change cannot explicitly tell us which ECU has been compromised, it does tell us that a system compromise has taken place.

Factor: Frequency

4.3.4 Bus Load Increase

As discussed under the Frame Frequency IOC, a significant change in the frame frequency can be detected by most IDSs. However, other methods exist to increase the frequency of frames slightly so that IDSs designed for such a purpose cannot detect the changes. As a result, the overall bus load will increase, causing the bus not to be able to transmit more messages which results in a Denial of Service (DoS) attack.

In order to perform this type of attacks, the assumption is that the attacker takes control over an ECU, especially the ECUs which transmit lots of different message IDs such as a gateway which connects two subnets to each other. If the attacker simply injects a message anywhere between two legitimate messages, the IDS will detect it because the time interval will significantly be shortened. Instead, the attacker must prevent the ECU from sending the legitimate message after injecting an own message. As figure 4.2 shows, the message injection is happening only 2 ms less than the legitimate frame periods which can be within the range of expected period. In 1 sec, it adds only 0.83 extra message to the system, but if it happens at a large scale e.g., 1000 message types at a gateway, the result can be 830 more messages per second which is a significant number.



(a) Normal CAN Messages

(b) Injected CAN Messages

Figure 4.2: Smart Message Injection

Every vehicular network technology has a maximum bit rate capacity. As an example, the maximum data rate for the standard CAN is 1 Mbps which is one million bits per second. The size of a standard CAN message is 128 bits (at max) plus 3 bits of interframe space. Thus, CAN is capable of transmitting roughly 7,700 messages per second. However, J1939 which is the dominant higher layer protocol for heavy-duty vehicles only supports 250 Kbps, and the most recent version is upgraded to support 500 Kbps. So, a J1939 network can only transmit roughly around 3,800 messages per second. These numbers are the maximum thresholds for CAN and J1939 networks. Exceeding these numbers will cause a DoS. Though, in reality, these many messages are not transmitted over vehicle networks, and it entirely depends on the manufacturers and the vehicle models, that at most, under different driving conditions, how many messages are transmitted. Such numbers can be found by driving the vehicle in different conditions and monitoring the network for the number of messages per second. Then, the obtained numbers can be used as an average threshold, which if exceeded, we consider it as an IOC.

Factor: Frequency

4.3.5 Existence of Conflicting Frames

While some message injections might not have a severe consequence on the safety of the vehicle, many of them do. For example, the message indicating the status of the door can be sent every 2 seconds [13]. An adversary can inject a message indicating the door is ajar, but this message will be replaced the next time the legitimate message is sent by the door sensor indicating the door is closed. Such an injection does not have a severe safety impact. However, an injected message indicating a clockwise x angel wheel turn while the legitimate message indicating anti-clockwise x angel wheel turn can have a severe safety impact. For example, while Miller and Valasek demonstrated their attack on the steering ECU [13], one conflicting frame indicating the opposite of the legitimate frame, deactivated the steering function. Fortunately, there are some safety measures considered by vehicle manufacturers, e.g., the Intelligent Parking Assist accepts a message indicating a steering wheel turn of high degree (e.g., more than 90 degrees) only if the vehicle is in reverse gear [13]. Although this type of injection should be captured by the frame frequency detecting mechanisms, conflicting within a very short time period can be an IOC. However, in order to keep the false positives low, several messages are needed before an alert can be raised confidently.

Factor: Frequency

4.3.6 Spamming Request for ECUs

Request frames are very interesting for attackers since most of the attacks can only be performed using such frames. As [13] explains, identifying the frames which indicate the level of accelerator pressure and injecting them with a different payload does not increase or decrease the speed. Instead, the frames might be intended for another ECU to inform if a certain event is in progress. There exist several different types of requests such as diagnostic requests, or requests as specified by J1939 but discussion of them is beyond the scope of this thesis project.

Factor: Frequency

4.3.7 Presence of Unknown Frame IDs

As we already discussed in section 2.2, a node sends data with 11-bit or 29-bit frame identifier. This unique identifier not only represents the message priority in the arbitration process but also describes the meaning of the data. Due to the nature of CAN, all the messages transmitted by the ECUs are broadcast to the network, and the receivers decide to process the message or not based on the ID. For the standard CAN, the frame ID can be in the range of 0 to 2047, but in reality, which frame IDs are being used can be different depending on the vehicles and the architectures. For example, one manufacturers may use only 50% (1024) of the available frame IDs for data transmission. In the case of extended CAN, the number of available IDs is 2^{29} which makes a total of more than 536 million possible IDs. Since the frame IDs are also used during the arbitration process, attackers are mostly interested in sending the lower ID frames in order to win the arbitration and override other frames. However, such IDs might no be used in that specific architecture; hence making the frame ID unknown or not expected. An IDS implemented at the network level can see all the traffic and flag upon the presence of such unknown or unexpected IDs. Although the protocols allow 2^{11} or 2^{29} possible IDs, and in a normal case not all of are being used, presence of expected IDs is a violation of the protocol.

Factor: Protocol

4.3.8 Sudden Changes in Signal Values

Most of the events on signal level are interrelated with each other. An event at time $t-1$ must have happened so that the relevant event at time t can happen. For example, if at time $t-1$ we observe a speed of 10 km/h, we do not expect a speed of 30 km/h at time t which in most cases is just a few ms later, considering the periodicity of messages. For example, as demonstrated in [37], while the vehicle is standing still, the attackers inject messages indicating that the vehicle is moving at a speed of 40 MPH. Due to the periodicity nature of the messages, the injected messages are being replaced by the legitimate messages and the speedometer is rapidly changing between 0 and 40. However, when the change is not huge the detection becomes very challenging. As discussed in [31], the forged speed which was only 1 km/h more than the actual speed could not be detected by their IDS; since this change was not an unexpected sudden change and was part of the normal behavior of the vehicle. However, since the sudden changes are mostly done by a means of message injection, it should be detected by the Frame Frequency factor.

Factor: Plausibility

4.3.9 Unexpected Diagnostic Messages

There are several different types of CAN messages such as diagnostics messages, messages with control signals, messages with sensor signals, and messages with status signals. In this section, we focus more on the diagnostic CAN messages. In order to check the state of health of a vehicle, update the ECU firmware or patch some known issues, manufacturers use several different protocols for vehicle diagnostic purposes. Keyword Protocol 2000 (KWP2000), Unified Diagnostic Services (UDS), Universal Measurement and Calibration Protocol (XCP) are some of the standardized protocols. Moreover, there are research works proposing other protocols to be used for diagnostic purposes as well, such as [38] [39] [40]. One type of diagnostics is a read-only diagnostic which only reads the Diagnostic Trouble Codes (DTC) to trace the faults in a vehicle. The read-only diagnostic is not a big issue from the security point of view. However the problem here is that this is a client - server setup, so the client actively needs to request the information, i.e. write to the bus, so a compromised diagnostics client could write arbitrary messages to the bus (unless firewalled) and arbitrary writes can be harmful. Several cheap commercial products are available in the market, especially for passenger cars, which reads out the DTCs and give some general instructions to the owner. However, the second type of diagnostic executes commands which alters ECU code and potentially the behavior of the vehicle [40]. This type of diagnostic can be dangerous from a security point of view because if an attacker successfully establishes such a diagnostic session with the vehicle, it is possible to alter ECU code.

Usually, the interface used for diagnostic purposes is OBD-II port which is now mandatory for all vehicles sold in the US and Europe. However, Miller and Valasek demonstrated that this could also be done remotely by connecting to the Telematics Unit [12]. Although manufacturers are using access control mechanisms to allow only authorized diagnostic clients to connect to the vehicle, the security of such mechanisms is quite fragile. Herrewegen et al. have successfully bypassed the authentication mechanisms of several famous vehicle manufacturers [41]. The authors mentioned the ciphers used for such authentications are usually 24 or 16 bits only. The paper was presented during the Vehicle Electronics & Connected Services 2019 conference in Gothenburg, Sweden and it was mentioned that it took them only a few hours to exhaust search the ciphers and break them. Thus, from a security point of view, we cannot yet fully rely on such authentication mechanisms to prevent attackers from tampering the vehicle ECUs. Additionally, based on the nature of CAN, messages with lower identifiers have higher priority over the messages which have higher identifiers. Herrewegen et al. mention that diagnostic CAN identifiers are usually between 0x700 and 0x7FF, but this is manufacturer specific [41]. All manufacturers have a list of legitimate diagnostic messages that they expect to see over the network.

As a safety standard, vehicle manufacturers usually perform the diagnostics only when the vehicle is in a Safe state. The Safe state can have different definitions by different manufacturers. One definition could be that the vehicle is parked with the parking brake engaged or maybe even the vehicle must be switched off with the parking brake engaged to be considered in Safe state. Regardless of which diagnostic protocol is being used, most of the diagnostic messages should only appear on the network when the vehicle is in a Safe state. The existence of certain diagnostic messages such as writes, reprogramming, ECUReset are unexpected over the CAN network while the vehicle is not in the Safe state is an obvious IOC. However, redaing DTC is expected and in most cases can harmless.

Factor: Consistency

4.3.10 Unexpected Behavior in Different Modes

In order to perform an attack in a vehicle, the attacker usually exploits a vulnerability when the vehicle is running or at least when the ignition is on. Manufacturers put effort into securing the vehicle when the ignition is on, while there are also possible attacks that performed when the ignition is off. Kyong-Tak Cho et al. came up with two different types of attacks that can be performed when the ignition is off: battery-drain attack and Denial-of-Body-control (DoB) attack [42].

Most ECUs are in an off state when the vehicle is parked, e.g. braking control ECU and parking ECU, While some ECUs that are designed to enhance user experience that stays in a sleeping mode when the ignition is off, these ECUs can be woken up easily by some signals or some software bugs [43][42]. For instance, the Passive Keyless Entry (PKE) is a vehicle security system that allows users to automatically unlock the door when approaching or lock it when leaving.

Kyong-Tak Cho et al. successfully woke up the sleeping ECUs by injecting any basic/standard CAN bus message (11-bit IDs) and it is said by using similar methodologies, ECUs can also be waked up through sending extended CAN bus message (29-bit IDs), FlexRay data or LIN data [42]. By constantly waking up sleeping ECUs, illuminate internal/external lights, change vehicle's power mode, unlock/lock doors, turn on the climate system and pre-heater, etc., the battery consumption speeds up. While in a normal case, when the ignition is off, the battery drain is pretty slow, so if the battery resource depletion is abnormal; the adversary has high possibility to inject a message in the CAN bus and perform the battery drain attack. So if ECUs receive CAN bus message to deplete lots of battery when the ignition is off, then the vehicle is probably compromised.

Factor: Consistency

4.4 Processing Layer IOCs

At the lowest level of the architecture, we have the ECUs that perform the actual jobs. In the following section, we list several IOCs we found in the processing layer.

4.4.1 Unexpected High Resource Usage

Resource utilization in vehicles is very different than traditional computers. Depending on the usage of a computer, resource utilization varies a lot. For example, a computer being used for gaming purposes has much higher resource utilization in comparison to the one being used for running a word processing software. However, resource utilization in vehicles is mostly consistent.

4.4.2 Failed Input Validation

Failure to validate the input data could be a sign of malformed or tainted input data. In order to bypass the validations, the attackers usually need to attempt countless times to forge the input data, which also generates hundreds of failed validation system logs, those continuously failed input validation logs could be a sign of compromise. Furthermore, the stored event logs associated with efficient user contexts can be used to identify suspicious activities.

4.4.3 Control Flow Integrity Violation

Control Flow Integrity is known as CFI, it prevents various malware attacks from redirecting the program execution flow. Strictly speaking, it must check each indirect control transfer and ensure that each transfer instruction can only be transferred to its own target set. Also, the received control flow should be consistent as the source code designed control flow attributes. Although deploying a context-sensitive checking mechanism maximizes the security of the system, its overhead is too large to be practically deployed. Attackers can hijack the control flow and gain control of the target machine or perform rights-raising operations to fully control the target machine. Violation of control-flow integrity would also be a violation of memory safety. So if there is a violation of CFI, we can say it's an indicator of compromise.

4.4.4 Data Mismatch from Multiple Sources

A vehicle speed sensor (VSS) is a sensor used for reading the speed of a vehicle's wheel rotation and measure the speed of a vehicle's wheels. It provides the speed information for the dynamic control system (VDC), automotive electronic stability program (ESP), anti-lock braking system (ABS), automatic transmission control system, etc. Therefore, the wheel speed sensor is one of the most critical sensors in modern vehicles and it's become a target for attackers. By compromising the VSS, the attacker can provide fake wheel speed data to the controller to influence the operator in maintaining control of the vehicle. For instance, by reducing the displayed wheel speed, the driver might accelerate the vehicle without knowing (s)he is already exceeding the speed limit. A GPS system is a high precision satellite navigation system, based on the assumption of the GPS is not hacked, Serrano et. al [44] confirmed that the satellite GPS velocity predicted in the navigation message is sufficiently accurate. If there is a mismatch between the GPS velocity and the wheel based speed sensor, we can guess there is a hacker trying to compromise the system. So the data mismatch from multiple resources can be an indicator of compromise.

In modern vehicles, there are two different types of VSS using in the vehicle systems: magneto-electric wheel speed sensor and hall effect speed sensor [45]. Compare with the hall effect speed sensor, the magneto-electric wheel speed sensor frequency response is low. When the vehicle speed is too high, the frequency response of the sensor can't catch up, which makes it easy to generate false signals. So because of the devices, there can also be a data mismatch which results in false positives.

4.4.5 Request for non-existing Services

As we already explained in section 4.3.9, UDS is a diagnostic protocol used in the CAN application layer. It's a standard for diagnostic services based on ISO 14229. It specifies what instructions should be sent to ECU when reading the fault code, what instructions should be sent when reading the data stream and so on. There are 26 different types of services in the UDS, including Diagnostic and Communications Management, Data Transmission, Stored Data Transmission, Input/Output Control, Remote Activation of Routine, Upload/Download, etc. Service Identifier (SID) is used to identify all those diagnostic services. UDS is essentially an interactive communication protocol (Request/Response), the diagnostic party sends the specified request data with SID to the ECU, if it is a positive response, then the receiver replies [SID+0x40], that is, request 10, response 50; request 22, response 62. The reply is a set of data. If the response is negative, then the receiver replies [7F+SID+NRC]. The reply is a statement. As Table 4.1 shows, when there is a request associated with a SID from an ECU, a corresponding SID should be included in the response message. While if the receiver receives a non-existing SID, then the sender probably is an attacker, namely, the system is compromised.

Function Group	Request SID	Response SID	Service
Diagnostic and Communications Management	$10	$50	Diagnostic Session Control
	$11	$51	ECU Reset
	...	...	...
Data Transmission	$22	$62	Read Data By Identifier
	...	...	...
Input/Output Control	$2F	$6F	Input/Output Control By Identifier
...	...	...	...

Table 4.1: UDS Communication

4.4.6 Integrity Check Failure

There are two mechanisms in CAN to check the data integrity: Error Frame and CRC checksum. If an ECU receives a bit that shouldn't be in the frame, means the frame is corrupted, then an error frame will be sent out instead. An ECU can also corrupt its own frame if it receives a bit different from what it has written. A 15 bit CRC checksum mechanism is implemented in the current CAN protocol, and it can detect all single-bit errors. Even though these two mechanisms can detect most of the integrity errors, it still can not guarantee the integrity of messages sent over the bus since Error Frame and CRC are two weak safety measures. If there is no additional integrity checking implemented with a CAN checksum at the application level, an attacker can inject some data and bypass the CRC checking and perform an attack, so usually, application-level integrity checks are also needed. Even though error frame and CRC are not strong enough to detect all kinds of failures in the data processing layer, they still indicate the anomalies behaviors, so an integrity check failure can be an indicator of the system compromised by the attacker(s).

4.4.7 Verification Failure of Exchanged Information

In current Intelligent Transport Systems (ITS), the communication between a vehicle and infrastructure is secured by the communication protocol with Secure Sockets Layer (SSL) or Transport Layer Security (TLS) implemented. Both SSL and TLS use handshake mechanism to establish secret keys for the communication between the client and the server. Since the transmitted data is encrypted, this handshake protocol ensures the hacker cannot see the plaintext without decrypting the message with the private-public key pair. This encryption protected exchanged data from alteration by attackers, namely provides information integrity. However if the exchanged information integrity verification fails, it can be a man-in-the-middle attack, namely, the communication is compromised.

4.4.8 Secure Boot Failure

Attackers try to launch attacks against the embedded systems in vehicles, once they succeed in breaking into the system, they can sabotage vehicle systems by triggering faulty processes. By trying to compromise the embedded system, attackers bring hidden malware into vehicles, which causes an enormous threat to the system. Even though this types of attacks are not happen very often these days, still in order to protect against these attacks, a secure boot mechanism is implemented to verify the signature of all software before they execute. When the system boots, the firmware first checks the signature of each boot software, if it's a valid signature, the software boots, otherwise, attackers probably already tamper with the embedded vehicle system. Since OEM creates secure boot private key pairs, every time new software is installed in the system, a new signed key pair should be stored in the secure boot database. So if the secure boot failure happens, we can say it's an indicator of compromise.

5

IOC Qualification Criteria

In this chapter, we introduce several criteria as the basis for a qualitative evaluation of the IOCs. These criteria are divided into three primary aspects: 1) Contribution in Reducing False Positives, 2) Cost of Implementing an IOC, and 3) Cost for Attacker to Evade an IOC. The evaluation results of all aspects are then used to determine the quality level of each IOC. Furthermore, each of the aspects is broken down into more factors since a simple and straightforward determination is not feasible and it will not be accurate. For example, for evaluating the implementation cost of an IOC, we examine the requirements that must be fulfilled in order for the IOC to be successfully implemented on a system. Similarly, to evaluate the cost for an attacker to bypass an IOC, we use the Common Methodology for Information Technology Security Evaluation (CEM) [29] (ISO/IEC 18045) technique to determine the required level of expertise, the required knowledge about the target, and the required equipment that the attacker(s) must have. There are more criteria that are used to achieve the final evaluation result, and we explain them throughout this chapter.

The three criteria to evaluate the quality of an IOC are as follow:

- Reduce False Positives
- Implementation Cost
- Evading Cost

5.1 Reduce False Positives

As discussed earlier in this report, one of the major issues of the IDSs is the high number of false positives, and this issue has prevented them to be fully applicable in vehicular industry due to the safety-critical requirements and the huge amount of data being processed. Also, the principle idea behind this thesis is to make a contribution toward reducing the number of false positives of the IDSs. Therefore, how much contribution an IOC can make in reducing the number of false positives has significant importance during the evaluation process. Since a quantitative evaluation is not feasible in this context; in order to prevent the evaluation procedure from being subject to criticism, we define only two levels, namely *high* or *low*. However, it is possible to have several other values in between, but their determination at this point is not possible. Instead, it is more accurate to implement the IOCs and observe the number of false positives related to them. To determine if an IOC can have a high or low contribution in reducing the number of false positives, we try to answer the key question: Can this IOC be mistaken with a system component failure which is not an attack? If the answer is "Yes", then the IOC is has a *low* contribution, but if the answer is "No", then the contribution is *high*.

5.2 Implementation Cost

The required cost to implement an IOC is the second key aspect in the IOC evaluation. Obviously, the cost is an important and considerable factor in every industry, but the vehicular industry is very sensitive to this factor due to the high volume of production. The cost to implement an IOC is partially dependent on the approach that an implementer would take, but there exist a minimum requirement that must be fulfilled first. We have defined three levels for the implementation cost, namely *low*, *medium*, and *high*. To determine which level of cost is needed by an IOC to be implemented, we examine the minimum requirements which must be fulfilled in order for an IOC to be successfully implemented. The requirements are as follow:

- Architectural changes
- Hardware
- Algorithm
- Complex software
- Training
- Multiple nodes
- Software customization
- Off-the-shelf software

Our intention has been to not involve the amount of money as part of the evaluation, since that is entirely dependent on the financial strength of individual stakeholders or an organization as a whole. While X amount of money might be considered cheap for one organization, it could be considered expensive or even not affordable for another organization. In the context of this evaluation, we only determine the levels of cost, and later the organizations can map them to the amounts of money that they consider low, medium or high.

5.2.1 High

The cost to implement an IOC is considered *high* if:

- it requires that the existing network architecture of a vehicle must be changed, or
- new hardware must be installed

In addition to the above requirements, an implementation might also require that a new algorithm must be designed and be implemented as a complex software which the complex software, in turn, might require new hardware with more processing power. There should be several requirements, the one with the highest cost is selected. For example, the cost for implementation with all of the above requirements remains *high*.

5.2.2 Medium

The cost to implement an IOC is considered *medium* if:

- it requires that a new algorithm must be designed, and/or
- a complex software must be developed, and/or
- the models must be trained using machine-learning or AI techniques, and/or
- the number of nodes that the IOC must be implemented on is multiple

A combination of all of these requirements is still considered medium.

5.2.3 Low

The cost to implement an IOC is considered *low* if:

- it only requires that an off-the-shelf software is implemented in the system, and/or
- an existing software must be customized

A combination of all of these requirements is still considered low.

5.3 Evading Cost

As the third aspect toward the evaluation of an IOC, it is important to evaluate the cost for the attacker(s) to bypass the IOC or evade it. It is a significant aspect due to the fact that a higher cost for the attacker(s) would lower the chances of the system being attacked. In order to evaluate the Evading Cost (EC), we have used the Common Criteria (CC) defined in Common Criteria for Information Technology Security Evaluation (CEM) [46] (ISO/IEC 15408). CEM describes the use of the following five CC to evaluate the potential of an attack:

1. the time required to identify a vulnerability and exploiting it (elapsed time) [1]
2. the specialized technical expertise required to perform the attack (expertise)
3. the required knowledge about the target (knowledge about the target)
4. opportunity of access to the target (window of opportunity)
5. the required hardware or software equipment (equipment)

In the context of this thesis, the higher the requirements to launch an attack, the more costly the attack would be for the attacker(s). In other words, we could expect to see a fewer number of such attacks.

5.3.1 Elapsed Time

Elapsed time is the time taken for a single attacker to identify a potential vulnerability in the system, develop hacking methodologies and compromise the system. It should be the time taken in the worst scenario, namely the maximum time to perform the attack.

[1] From version 3.1, CC considers the identification and exploitation a vulnerability as one phase. Previously they were two distinct phases, and each of them had its own time requirement.

Short: Time cost is less than a day.

Medium Short: Time cost is less than a week.

Medium Long: Time cost is less than a month.

Long: Time cost is more than a month.

5.3.2 Specialist Expertise

The generic/specialist knowledge required to carry out an attack, including the knowledge about the product, the principles of the methodologies, attack approaches, etc. When the attack happens, the indicators can be seen.

Layman: Does not require any particular expertise. It can be the ordinary vehicle owner/driver who knows several simple attacks.

Proficient: Require general security knowledge. They can be some experienced owners or the ordinary garage personnel who is familiar with some security behaviors of the products. They probably can perform some attacks with some available tools and instructions.

Expert: Requires expert security and domain knowledge including algorithms, principles and protocols, software and hardware. They can be some experienced adversary who know some sophisticated published attacks and can perform new attacks.

Multiple Experts: Require expert security knowledge in different fields. They can be highly experienced personnel with state-of-the-art knowledge.

5.3.3 Knowledge about the Target

From which sources an attacker could get the information related to the target. It's distinct from the expertise since it represents the availability of the attack related information.

Public: The information about the target is public; everyone has access to it. Information is shared without non-disclosure agreements. For example, the information found over the Internet.

Restricted: The information about the target is shared among different partners under a non-disclosure agreement.

Sensitive: The information about the target is shared among different teams within the organization under a non-disclosure agreement. It's only accessible to the team members.

Critical: The information about the target is only accessible to a few individuals within the organization. The critical information has an extremely strict access process.

5.3.4 Window of Opportunity

Required physical or remote access time for an attacker to take advantage of the vulnerability and compromise the system. It has a relationship with the "Elapsed Time" since identification and exploitation might require access to the target. The access to the target reveals the attacker's activities, so the window of opportunity should also consider the number of targets the attacker needs.

Unlimited: Unlimited physical and remote access, namely the attacker can always access the target without being detected. In this case, the amount of access to the target doesn't affect the opportunity of being detected.

Easy: The required physical or remote access is less than a day and/or the number of target samples needed to perform the attack is less than 10.

Moderate: The required physical or remote access is less than a month and/or the number of target samples needed to perform the attack is less than 100.

Difficult: The required physical or remote access is more than a month and/or and the number of target samples needed to perform the attack is more than 100.

5.3.5 Equipment

Required software or hardware for taking advantage of the vulnerability and compromising the vehicular system.

Standard: The required software and hardware is readily available to everyone.

Specialized: The equipment is not readily available, but can be obtained without excessive effort.

Bespoke: The attacker needs some specially produced equipment or multiple types of specialized equipment.

Multiple Bespoke: Multiple types of bespoke equipment are required to perform the attack.

5.3.6 Evading Cost Parameters

We only use three of the above criteria namely *expertise*, *knowledge about the target*, and *equipment*. The reason is the strong dependability among the criteria. The *elapsed time* is entirely dependent on the criteria that we are going to use. For example, to perform an attack, it might take an expert with multiple bespoke types of equipment only one day, but performing the same attack might take a professional with specialized equipment more than a week. *Window of opportunity* is also not fully relevant to our thesis since we assume that the attacker(s) could buy an instance of the vehicle as their target, so it is always available to them. For launching remote attacks, they could obtain the required access if they get within the coverage area of the remote interfaces. It is worth mentioning that we have provided a brief description for all of the five criteria to give the reader an insight on what the criteria are and to justify the reason of not using them as part of our evaluation.

Wolf et al. [47] and Islam et al. [48] have also used the above factors in their two-dimensional security risk assessment frameworks to calculate the *attack potential* and *threat level* respectively. While the former uses all of the five criteria, the latter only uses four of them.

The parameters in the Table 5.1 are taken from the CEM [B 4.2.3] with minor modifications. For example, the values for each parameter are chosen in the range of 0 - 3. There are two reasons for making this modification. The first reason is lack of justification for the chosen values in the standard as well as [47]. Although lack of justification may not reduce the validity of the methodology, it might have considered some aspects that might not be feasible for this thesis. The second reason is that we have considered a *weight* for each criterion, but we leave it to the user of the framework to choose a proper value for the weight.

Specialist Expertise		Knowledge about the Target		Equipment	
Layman	0	Public	0	Standard	0
Proficient	1	Restricted	1	Specialized	1
Expert	2	Sensitive	2	Bespoke	2
Multiple Expert	3	Critical	3	Multiple Bespoke	3

Table 5.1: Evading Cost Parameters

5.3.7 Evading Cost Classification

In order to evaluate EC, a value from table 5.1 should be assigned to each sub-criterion. To obtain the result, the following linear equation can be used.

$$EC = W_x X + W_k K + W_e E$$

Where X is Expertise; K is Knowledge about the Target; E is Equipment; W_x, W_k, and W_e are the weight for each criterion. We have set the *weight* equal to 1, but the user of the framework has the flexibility to adjust it according to their requirements. A classification of the EC is shown in table 5.2.

EC Value	EC Classification
>4	High
3-4	Medium
0-2	Low

Table 5.2: Evading Cost Classification

5.4 IOC Quality Classification

In order to determine the quality of an IOC, we have considered the following three aspects:

- Reducing False Positives (RFP)
- Implementation Cost (IC)
- Evading Cost (EC)

Each of the above criterion should be evaluated using the methodologies defined in this chapter. In order to determine the quality level of an IOC, the below three-dimensional table should be used (see table 5.3). The IOC with the highest quality should have the most significant contribution in reducing the number of false positives, it should require a low implemented cost, and at the same time, it should have have a high cost for the attacker(s) to evade it. Similarly, the IOC with the lowest quality should be the one which can have a low contribution in reducing the number of false positives, it should require a high implementation cost, and the same time it should have a low cost for the attacker(s) to evade it. In table 5.3, 1 is the lowest level and 3 is the highest level. The IOCs with attributes in between the above mentioned quality attributes are classified to the closest level.

			EC		
		IC	Low	Medium	High
RFP	Low	High	1	1	1
		Medium	1	1	2
		Low	1	2	3
	High	High	1	2	2
		Medium	2	3	3
		Low	2	3	3

Table 5.3: IOC Quality Classification

6

Results

The IOCs presented in chapter 4 are evaluated in accordance with the criteria and the methodologies defined in chapter 5, and the results of all evaluations are presented in this chapter. Section 6.1 provides the list of all IOCs found during this thesis project. Section 6.2 provides the result of evaluating the IOCs for the amount of contribution they can make toward reducing the number of false positives. Section 6.3 provides the result of evaluating the implementation cost of IOCs. Section 6.4 provides the result of evaluating the cost for the attacker(s) to evade the IOCs. Section 6.5 presents the results of combining all the criteria and evaluating the quality of the IOCs. Finally, in section 6.6 we propose a distributed IDS having sensors at four layers of an automotive network. We also propose a logic for the central IDS node to combine two or more IOCs in case a single IOC cannot achieve the level of confidence required to raise an alarm.

It is worth mentioning that the results presented in this chapter are all qualitative. A quantitative assessment in the subject of cyber-security is not feasible due to lack of required and crucial data such as a list of all possible attacks against an asset.

6.1 List of IOCs

In this section, we provide a summarized list of all IOCs found during this thesis project. The list is shown in table 6.1. The numbers just act as an identification mean, and have nothing to do with the ranking of the IOCs. These numbers are used in later tables for the sake of compactness. The only ordering in the table is the layers, starting from the external layer (interface) and moving down toward the processing layer (ECUs).

Layer	IOC_s	IOC Name
Interface	IOC_1	Port Scanning of the Connectivity Gateway
	IOC_2	Evil Twin SSID Existence
	IOC_3	TCU New MAC Address Connection
	IOC_4	Link Downgrade
Gateway	IOC_5	Information Presence on an Unexpected Network
	IOC_6	Integrity Failure in Routing Rules
Network	IOC_7	Unexpected Physical Characteristics of Signals
	IOC_8	Out-of-Range Signal Values
	IOC_9	Frame Frequency Increase
	IOC_{10}	Bus Load Increase
	IOC_{11}	Existence of Conflicting Frames
	IOC_{12}	Spamming Request for ECUs
	IOC_{13}	Presence of Unknown Frame IDs
	IOC_{14}	Sudden Changes in Signal Values
	IOC_{15}	Unexpected Diagnostic Messages
	IOC_{16}	Unexpected Behavior in Different Modes
Processing	IOC_{17}	Unexpected High Resource Usage
	IOC_{18}	Failed Input Validation
	IOC_{19}	Control Flow Integrity Violation
	IOC_{20}	Data Mismatch from Multiple Sources
	IOC_{21}	Request for non-existing Services
	IOC_{22}	Integrity Check Failure
	IOC_{23}	Verification Failure of Exchanged Information
	IOC_{24}	Secure Boot Failure

Table 6.1: List of IOCs

6.2 Reduce False Positives Evaluation Results

By answering the key question of "can this IOC be mistaken with a system component fault?", we evaluated all of the IOCs presented in table 6.1. While it is obvious that some of the IOCs cannot be the result of a system component fault, some others are not that straightforward. To bring more accuracy in the evaluation, we asked for help from a few industry experts in the field of vehicular security. The results are shown in table 6.2.

The IOCs that cannot be mistaken with a system component fault can be classified as independent IOCs and we have categorized them with High RFP. The independent IOCs are those which, if implemented, could detect an attack or a system compromise with a high degree of confidence alone. The dependent category contains those IOCs which need to be combined with one or more IOCs in order to achieve the desired degree of confidence and in the table there have RFP Medium or Low. An example of an independent IOC is the existence of special diagnostic frames (reprogramming of an ECU) during an unexpected state, e.g., while the vehicle is moving. Since we do not expect to see such diagnostic routines when the vehicle is not in a safe mode, seeing one of such frames would be enough to raise an alarm. On the other hand, the failure in integrity checking of a frame cannot be an independent IOC since it could be the result of a simple bit flip during the transmission. Additionally, the classification of the IOCs can be different depending on the configuration of a system. For example, when a TCU connects to new MAC address, this IOC can or cannot not be classified as an independent IOC. This IOC is not enough to raise an alarm if it is the result of replacing the head unit and forgetting to update the TCU with the MAC address of the new head unit. However, if there was a mechanism in place to guarantee that the TCU would not connect to a new head unit unless the MAC address is updated in its firmware, then this IOC would be enough to raise an alarm, hence it could be classified as an independent IOC.

IOC_s	RFP	IOC_s	RFP	IOC_s	RFP	IOC_s	RFP	IOC_s	RFP
IOC_1	H	IOC_2	H	IOC_3	H	IOC_4	L	IOC_5	H
IOC_6	H	IOC_7	H	IOC_8	L	IOC_9	L	IOC_{10}	L
IOC_{11}	L	IOC_{12}	L	IOC_{13}	H	IOC_{14}	L	IOC_{15}	H
IOC_{16}	H	IOC_{17}	L	IOC_{18}	L	IOC_{19}	L	IOC_{20}	L
IOC_{21}	H	IOC_{22}	L	IOC_{23}	L	IOC_{24}	L		

Table 6.2: Reduce False Positives Evaluation Results

6.3 Implementation Cost Evaluation Results

Similar to the evaluation procedure for the first criterion, we asked for help from the same industry experts to evaluate the implementation cost of all IOCs based on the criteria defined in this thesis. While the approach to implement some of the IOCs is known, some other IOCs can be complicated. For example, we know that detection of port scanning activity can be done using some open-source software that might need some minor customization. However, implementing the IOC which checks for the plausibility of interrelated events is not that straightforward. Evaluation result of all IOCs is shown in table 6.3.

IOC_s	IC	IOC_s	IC	IOC_s	IC	IOC_s	IC	IOC_s	IC
IOC_1	L	IOC_2	L	IOC_3	L	IOC_4	L	IOC_5	M
IOC_6	M	IOC_7	L	IOC_8	L	IOC_9	M	IOC_{10}	M
IOC_{11}	M	IOC_{12}	M	IOC_{13}	L	IOC_{14}	L	IOC_{15}	L
IOC_{16}	M	IOC_{17}	M	IOC_{18}	M	IOC_{19}	M	IOC_{20}	M
IOC_{21}	M	IOC_{22}	M	IOC_{23}	M	IOC_{24}	M		

Table 6.3: Implementation Cost Evaluation Results

6.4 Evading Cost Evaluation Results

Evaluation of this criterion is more challenging than the first two, due to the fact that we have to do it from an attacker perspective. Even though we do not know how an attacker would perform an attack to evade the IOCs, and in fact, that is not our intention, we still know that the IOC is able to detect the behavioral changes that the known attack would make in the system. So the attacker(s) need to gain a higher level of expertise, obtain more knowledge about the target, and even they might be required to develop their own sophisticated hardware or software. We do not claim that the IOCs presented in this thesis cannot be bypassed by any means, but we are confident that bypassing these IOCs would have a cost for the attacker, and we have tried to estimate that cost. Therefore, the accuracy of the evaluation is still reasonable. We again asked the industry experts to help us conduct the evaluation in order to have higher accuracy for the evaluation. Table 6.4 shows the results.

Legend: The character C in the column headers stands for criterion, C_1 is the criterion 1 namely "the required level of expertise", C_2 is criterion 2 namely "the required knowledge about the target", and C_3 is criterion 3 namely "the required equipment". See the list below:

- C_1 - Level of expertise
- C_2 - Knowledge about the target
- C_3 - Equipment

IOC_s	C_1	C_2	C_3	Sum	EC	IOC_s	C_1	C_2	C_3	Sum	EC
IOC_1	2	2	2	6	H	IOC_2	1	0	1	2	L
IOC_3	2	0	0	2	L	IOC_4	2	0	0	2	L
IOC_5	2	2	0	4	M	IOC_6	2	2	1	5	H
IOC_7	1	1	0	2	L	IOC_8	2	2	2	6	H
IOC_9	1	0	0	1	L	IOC_{10}	2	1	0	3	M
IOC_{11}	2	2	2	6	H	IOC_{12}	1	0	0	1	L
IOC_{13}	2	2	2	6	H	IOC_{14}	1	0	0	1	L
IOC_{15}	2	1	0	3	M	IOC_{16}	2	1	0	3	M
IOC_{17}	2	1	1	4	M	IOC_{18}	2	1	2	5	H
IOC_{19}	2	2	1	5	H	IOC_{20}	2	1	1	4	M
IOC_{21}	2	2	2	6	H	IOC_{22}	2	2	1	5	H
IOC_{23}	2	2	2	6	H	IOC_{24}	2	2	2	6	H

Table 6.4: Evading Cost Evaluation Results

6.5 IOC Quality Evaluation Results

After evaluating the IOCs with respect to all of the three criteria, a combination of all results is used to determine the quality of the IOCs. All other combinations between these two qualities are shown in table 6.5.

IOC_s	RFP	IC	EC	Quality	IOC_s	RFP	IC	EC	Quality
IOC_1	H	L	H	L_3	IOC_2	H	L	L	L_2
IOC_3	H	L	L	L_2	IOC_4	L	L	L	L_1
IOC_5	H	M	M	L_3	IOC_6	H	M	H	L_3
IOC_7	H	L	L	L_2	IOC_8	L	L	H	L_3
IOC_9	L	M	L	L_1	IOC_{10}	L	M	M	L_1
IOC_{11}	L	M	H	L_2	IOC_{12}	L	M	L	L_1
IOC_{13}	H	L	H	L_3	IOC_{14}	L	L	L	L_1
IOC_{15}	H	L	M	L_3	IOC_{16}	H	M	M	L_3
IOC_{17}	L	M	M	L_1	IOC_{18}	L	M	H	L_2
IOC_{19}	L	M	H	L_2	IOC_{20}	L	M	M	L_1
IOC_{21}	H	M	H	L_3	IOC_{22}	L	M	H	L_2
IOC_{23}	L	M	H	L_2	IOC_{24}	L	M	H	L_2

Table 6.5: IOC Quality Evaluation Results

By doing the evaluation, we get the result of:

- L_1: IOC_9, IOC_{10}, IOC_{12}, IOC_{14}, IOC_{17}, IOC_{20}
- L_2: IOC_2, IOC_3, IOC_4, IOC_7, IOC_{11}, IOC_{18}, IOC_{19}, IOC_{22}, IOC_{23}, IOC_{24}
- L_3: IOC_1, IOC_5, IOC_6, IOC_8, IOC_{13}, IOC_{15}, IOC_{16}, IOC_{21}

By considering "Reduce False Positives", "Implementation Cost" and "Evading Cost", we get a conclusion with three levels of IOC quality, which for the sake of simplicity, we call them as L_1, L_2, and L_3. Even though by considering all three criteria, L_3 IOCs are better than L_1 IOCs, it is important to understand that the L_1 IOCs still have a significant role.

6.6 IOC Deployment

In practice, an IOC can only protect a system against malicious attacks when it is implemented and deployed on the system. However, as shown in section 6.5, some IOCs if implemented alone, can have a low contribution in reducing the false positives since they can be a result of system malfunction. For example, an "integrity check failure" might be the result of a system bit flip during transmission but such failure would cause the corresponding IOC to indicate an attack. Therefore dependent IOCs need to be combined with at least one or more IOCs to achieve the desired level of confidence for raising an alarm. In order to avoid increasing the number of false positives, the logic for combining the dependent IOCs must be designed very carefully. As part of the logic design, finding the relationships among the IOCs is the most important factor.

We propose a distributed IDS as shown in figure 6.2. Four types of sensors should be placed, one at each layer namely, Interfaces, Gateways, Networks, and Hosts. If any of the sensors observe an anomaly which results in an IOC triggers, the incident must be reported to the central node. For the central node to be able to find the relevant IOCs and properly combine them to increase the degree of confidence, it should consider two important factors, 1) the order in which the IOCs trigger, and 2) the distance between the layers in which the IOCs trigger. We consider the second factor as distance between the IOCs. However, all possible combinations of the IOCs are not valid and cannot be part of the same attack. For example, if an IOC indicates a "message injection" triggers first, and then another IOC indicating "port scanning" triggers, the probability of these two IOCs being as part of the same attack is very low, since "port scanning" should happen earlier than "message injection". Similarly, the smaller the distance between the IOCs, the higher the probability of the IOCs being as part of the same attack. For example, if an IOC indicating the existence of an "evil twin SSID" triggers at the Interface layer, and consequently the second IOC indicating a "new MAC address connection" also triggers at the Interface layer, the probability of these two IOCs being as part of the same attack is much higher than a different second IOC triggers from the Network layer. There are four possible distances among the IOCs follow:

- 0: zero distance. IOCs are at the same layer, or the distance between the last IOC in one layer, and the first IOC at the next immediate layer.
- 1: one layer distance, e.g., between Interface and Gateway
- 2: two layers distance, e.g., between Interface and Network
- 3: three layers distance, e.g., between Interface and Processing

Since the order of IOCs matter, equation 6.1 can be used to determine all possible valid combinations of the dependent IOCs. In the equation n is the number of dependent IOCs, and m is the number of required IOCs to trigger so that the desired level of confidence is achieved. The higher the value of m, the higher the number of valid combinations. However, this fact is only true when $m <= n/2$. After $m <= n/2$, the number of combinations start repeating. It is also very important that the m is not selected very high, since doing so while increasing the degree of confidence also increases the number of false negatives. As a result, the sensitivity of the system could dramatically be decreased.

$$C(n, m) = \frac{n!}{m!(n-m)!} \tag{6.1}$$

The distance between IOCs is calculated as $[deeper\_layer] - [outer\_layer]$. For example, the distance between B and D is $3 - 1 = 2$. Also, the layers are ordered from outermost layer to the deepest layer in the system. The IOCs belong to different layers as follow:

- Interface (1): A, B
- Gateway (2): C
- Network (3): D
- Processing (4): E

In the example below, we determine all possible combinations of five IOCs and consider the distance between the two IOCs as well. Let's assume we have five IOCs namely, A, B, C, D, E. We also expect three IOCs to trigger so that the desired level of confidence is achieved. Applying the above formula, we get 10 valid combinations as follow.

$[A_0B_1C]$, $[A_0B_2D]$, $[A_0B_3E]$, $[A_2C_1D]$, $[A_2C_2E]$, $[A_3D_1E]$,
$[B_1C_1D]$, $[B_1C_2E]$, $[B_2D_1E]$,
$[C_1D_1E]$

It is worth mentioning that when an IOC triggers, the subsequent IOCs can only be selected from the list of IOCs with a higher order. For example, when B is selected as the first IOC to trigger, then only C, D, and E can be selected next, but not A. As can be seen in the list, the first two combinations, $[A_0B_1C]$ and $[A_0B_2D]$ are different in the sense that the distance between the second and the third IOCs are different. The distance value can be treated as a *probability* that the two IOCs are related, with closer distance being considered as higher probability, the first combination has higher degree of confidence in comparison to the second combination.

In order to perform an attack, there exist several attack paths for the attacker. Figure 6.1 shows only two, with the IOCs that should trigger along one of the paths. In this example, a valid combination of the IOCs is $[IOC_5 \text{ –> } IOC_6]$, but the reverse order is not a valid or reasonable combination. In order to find more valid combinations, more attack paths are required which, unfortunately, we are missing such data. Therefore, we leave this as future work. Similarly, defining the accurate probabilities among the interfaces requires more data such as real or virtualized test results. Also, we leave it to the user of framework to make a decision on selecting a proper value for m.

Finally, we also propose that the incidents that do not comply with the desired degree of confidence must be reported to the back office for further investigation. A very significant assumption here is that the distributed IDS is not compromised, and the connection to the back office is secure.

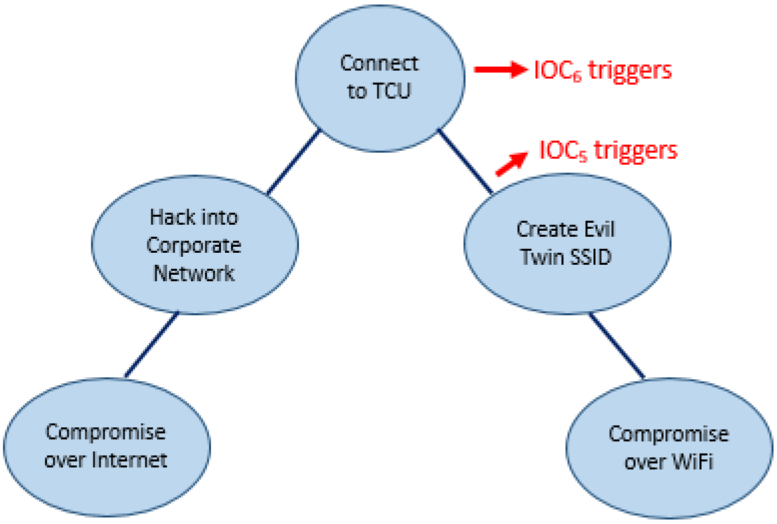


Figure 6.1: Remote Attack on TCU

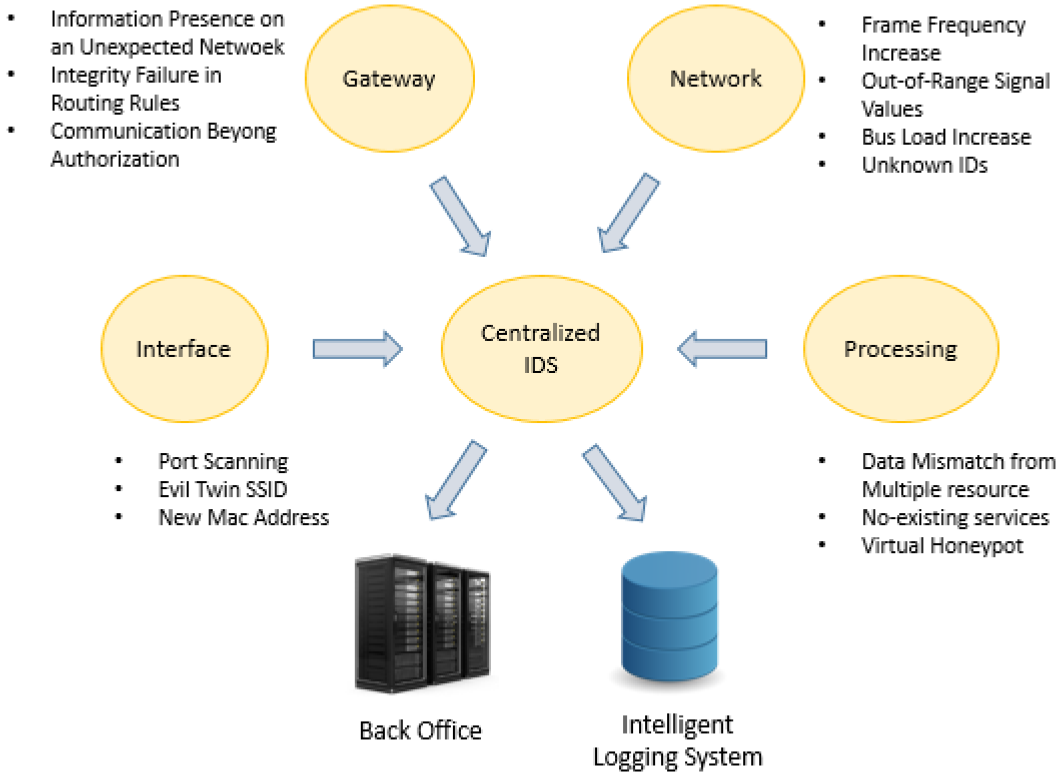

Information Presence on an Unexpected Netwoek
Integrity Failure in Routing Rules
Communication Beyong Authorization
Gateway
Network
Frame Frequency Increase
Out-of-Range Signal Values
Bus Load Increase
Unknown IDs
Interface
Centralized IDS
Processing
Port Scanning
Evil Twin SSID
New Mac Address
Data Mismatch from Multiple resource
No-existing services
Virtual Honeypot
Back Office
Intelligent Logging System

Figure 6.2: IOC Deployment

7

Discussion

Since a quantitative methodology was not feasible to conduct this research, we had to use a qualitative method. Additionally, since any qualitative piece of work can be subject to discussion, our research is not an exception. In this chapter, we discuss our thesis in general and try to cover all aspects that could potentially raise a question for the reader.

7.1 List of IOCs

The IOCs provided in this report are not the only possible IOCs in a vehicular system. Limitation of time is the most important factor in limiting the number of IOCs found and presented. However, while researching for IOCs, we have focused on the behavioral changes which an attack can make in a vehicular system. For instance, an increase in frame frequency of the network could be a result of dozens of different attacks, so by looking for this IOC in a vehicular system, dozens of attacks can be prevented.

Regarding the methodology of finding IOCs, a study of published attacks might not be the best possible way but as mentioned earlier, we have covered a substantial amount of footprints that a successful attack may leave in the system. Another methodology could be to implement machine-learning mechanisms and to assess the legitimate behavior of a vehicle.

7.2 Reduce False Positives

The evaluation of this criterion has also been performed in a qualitative approach. We, with the help of the vehicular security experts, tried to think of any system failure that might be mistaken with an attack. Where we could not find such a system fault, we have rated an IOC as it cannot be mistaken with any fault. However, after the implementation and examining a few test cases, the evaluation results can be subject of being changed. Since an IDS contains computer code, and in some cases it could be very complex, and due to the fact that software is subject to bugs, an IOC could also be mistriggered as a result of such bugs. Additionally, testing is a substantial phase in software development lifecycle, thus we considered this sub-criterion as a negligible factor and did not include it in the evaluation process. Also, it is very difficult to assign an accurate value to this factor since we do not know the accuracy rate of developing such systems.

Nonetheless, if the rate of an IOC for this criterion is changed after a more accurate and realistic tests, the framework can still be used to update the quality of the IOC in the final result.

7.3 Implementation Cost

To evaluate the implementation cost of an IOC, we considered the minimum requirements. In addition to the defined requirements, organizations could also consider other factors such as human resources and time. For example, the cost for an organization which has the human resources required to design a new algorithm or develop a complex software in-house would significantly be reduced in comparison to another organization which needs to outsource the tasks. Time can also be considered important in cases when an organization has a deadline to launch a product. For example, if there is a case which an IOC can be implemented by either installing new hardware or designing a new algorithm, naively thinking, the algorithm design seems to have a lower cost. However, it might take six months to be designed, but the hardware could be installed immediately. In such a case, where time matters, the organization might not have the option to opt the algorithm design.

7.4 Evading Cost

While evaluating the evading cost, motivation has not been considered as part of the evaluation. Even though the motivation could be regarded as the most important criterion, it is very difficult to be measured. In fact, the motivation could define everything in the context of attacking an asset. For example, a very motivated adversary would spend as much time and budget as needed and would acquire the required equipment and knowledge in order to succeed. Due to the difficulty of measurement, we decided to avoid involving the motivation in our evaluation process.

Additionally, during the evaluation process, we assigned the values to each criterion to the best of our knowledge, with the help of the vehicular security experts, regarding the performance of such attacks using the existing tools and technologies that we have studied. Nonetheless, the framework provided here is more important than the values, since the results are subject to changes. For example, if today an attack takes a tremendous amount of time to perform and requires advanced and sophisticated equipment, the same attack will become cheaper to perform after some time. For example, Miller and Valasek [13] published all of their research results along with the software they developed to perform the attacks ready to download at no cost.

7.5 IOC Qualification

Before defining the current qualification attributes, we also tried to use the detection rate and number of false positives that an IOC might generate. However, we found that such criteria are more relevant to the detailed implementation of an IOC instead of the IOC quality. Similar to the evaluation of evading cost, the classification of the same IOCs can vary if the evaluation is performed by a different organization due to the different interpretation of the attributes. While one organization might consider the implementation cost as high, another organization might consider it low, based on their financial strength, resulting in a higher quality for the IOC.

As another criterion to evaluate the quality of an IOC, one can use the coverage area of the IOC. The bigger the coverage area of an IOC, the more worthy to invest on the IOC. If, a quality evaluation is conducted and two IOCs have the same results, the IOC which can cover more area should be regarded as the better IOC. For example, while a network IOC can help in detecting the attacks for an entire subnet, the processing layer IOC can only help in detecting the attacks to/from one single ECU. Obviously, the network IOC has a better overall quality.

7.6 IOC Deployment

In order to be able to combine the dependent IOCs in a systematic fashion so to achieve the required degree of confidence for raising an alarm, a machine-learning mechanism should be implemented to draw all possible attack trees and examine the IOCs which will trigger along the attack paths from the outermost layer to the deepest layer in the system. By using a machine-learning mechanism, we could also find a range of time gaps between two IOCs and use such values to define a relationship between them. For example, if we could determine that when IOC_1 triggers, then IOC_2 will trigger after X amount of time, and then it will take another

X amount of time to see IOC_3 along an attack path, we could say, with a very high degree of confidence, that the three IOCs are related to each other, so they indicate the same attack. Lack of enough data to draw all possible attack trees and shortage of time prevented us from developing such a systematic framework for combining the dependent IOCs.

Additionally, not all combinations of the IOCs presented in this thesis project are valid. If two IOCs are seen at the same time in a system, it does not necessarily mean that they are related or they are part of the same attack. For example, if an IOC at the interface layer indicates that a port scanning is going on, and at the same time, another IOC indicates that a control flow violation happened at the processing layer, they are most likely unrelated to each other. The reason is that port scanning is usually the first step toward hacking into a system, since the attacker just tries to find an open port. In most of the cases, there might not be any open port to be used for getting into the system, but assuming that the attacker succeeds to hack into the system, we expect to see at least one more IOCs along the attack path toward the ECU which its control flow has been violated. Alternatively, there should be some time gap between the two incidences since starting an attack at the interface layer and reaching to the ECU, definitely requires some time, although the time might be very minimal. The two mentioned IOCs might still be related to each other, but for sure not when they happen at the same time.

Finally, if the triggering of two IOCs is considered enough for raising an alarm, then by having 13 dependent IOCs in total, the number of combinations will be 78. However, if three IOCs are required for raising an alarm, then the number of combinations will increase to 286. Finding all these combinations manually is very difficult, but it should not be very challenging by using machine-learning techniques and performing penetration testings in security laboratories.

8

Conclusion

Modern vehicles are equipped with plenty of sensors and Electrical and Electronic (E/E) systems such as Electronic Control Units (ECUs). The E/E systems are used to control the primary functions in vehicles such as engine control, body control, transmission and braking systems as well as safety functions such as airbag, Advanced Driving Assistance Systems (ADAS), e.g., adaptive cruise control. The number of ECUs in a modern vehicle goes beyond 100. In order to communicate with each other, ECUs require to have a uniquely designed network to support the safety-critical functions. Such networks must be capable of performing in real time and must have bounded delays. Among many such networks, Controller Area Network (CAN), has been widely accepted by vehicle manufactures due to its low cost of implementation and its bounded delay characteristic. However, CAN was designed without having the potential cyber-security threats in mind. J1939 is a higher layer protocol with CAN as the basis, specially designed for industrial vehicles such as truck and buses. Moreover, in order to program the ECUs, hundreds of millions of lines of code are written and such programs are not bug-free. As a result of having a complex software, complicated internal network and connecting to the external world, vehicles have evolved into drivable computers. Similar to traditional computers, modern vehicles also face computer and network threats. Both intruders and researchers have compromised the security of vehicles in different ways either locally or wirelessly. Just as an example we can mention the attack performed by a team of researchers, led by Prof. Stefan Savage from the University of California, San Diego, and Tadayoshi Kohno from the University of Washington that experimentally evaluated the security issues of a modern vehicle and demonstrated that the underlying system structure is quite fragile. Hacking the Jeep by Miller and Valasek in 2015 was the headline of tech news for a while.

To protect the vehicles, against the cyber-attacks, several security mechanisms are proposed, within which the Intrusion Detection Systems (IDS) have gotten the most attention. However, the biggest issue of IDSs is the high number of false positives that they generate. A false positive is when a legitimate incident is reported as an attack. In this thesis project, we found several Indicators of Compromise (IOC) in order to mitigate the number of false positives. Additionally, we have proposed a set of criteria as well as methodologies to evaluate the quality of an IOC. Nonetheless, some of the IOCs presented in this thesis are strong indicators which if triggered, an alarm can be raised with a high degree of confidence, but some others need to be combined with one or more IOCs to achieve a desired degree of confidence. We have proposed a distributed IDS with sensors at several places in a vehicular system. The sensors should report the incidents to the central node to make a decision to either raise an alarm or not. We have also proposed logic for the central node to combine the IOCs which cannot achieve a desired degree of confidence to raise an alarm. When the desired degree of confidence is achieved, the central node can raise an alarm.

Bibliography

[1] Adam West. NASA Study on Flight Software Complexity. https://www.nasa.gov/sites/default/files/418878main_FSWC_Final_Report.pdf, 03 2009. [Accessed on: 13-March-2019].

[2] CAN Specification. http://esd.cs.ucr.edu/webres/can20.pdf. [Accessed on: 11-February-2019].

[3] Robert N Charette. This Car Runs on Code. *IEEE Spectrum*, 46(3):3, 2009.

[4] Ishtiaq Rouf, Rob Miller, Hossen Mustafa, Travis Taylor, Sangho Oh, Wenyuan Xu, Marco Gruteser, Wade Trappe, and Ivan Seskar. Security and Privacy Vulnerabilities of in-car Wireless Networks: A Tire Pressure Monitoring System Case Study. In *19th USENIX Security Symposium, Washington DC*, pages 11–13, 2010.

[5] Motor Vehicles Increasingly Vulnerable to Remote Exploits. https://www.ic3.gov/media/2016/160317.aspx, March 2016. [Accessed on: 16-April-2019].

[6] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462, May 2010.

[7] Hack Attacks Mounted on Car Control Systems. https://www.bbc.com/news/10119492, May 2010. [Accessed on: 11-February-2019].

[8] Rebecca Boyle. Proof-of-Concept CarShark Software Hacks Car Computers, Shutting Down Brakes, Engines, and More. Popular Science, https://www.popsci.com/cars/article/2010-05/researchers-hack-car-computers-shutting-down-brakes-engine-and-more, May 2010. [Accessed on: 01-March-2019].

[9] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *USENIX Security Symposium*, pages 77–92. San Francisco, 2011.

[10] Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage. Fast and Vulnerable: A Story of Telematic Failures. In *WOOT*, 2015.

[11] Jose Carlos Norte. Hacking Industrial Vehicles from the Internet. `http://jcarlosnorte.com/security/2016/03/06/hacking-tachographs-from-the-internets`, 03 2016. [Accessed on: 21-March-2019].

[12] Charlie Miller and Chris Valasek. Remote Exploitation of an Unaltered Passenger Vehicle. *Black Hat USA*, 2015.

[13] Charlie Miller and Chris Valasek. Adventures in Automotive Networks and Control Units. *Def Con*, 21:260–264, 2013.

[14] Charlie Miller and Chris Valasek. Jeep Hackers at it Again, This Time Taking Control of Steering and Braking Systems. The Verge, `https://www.theverge.com/2016/8/2/12353186/car-hack-jeep-cherokee-vulnerability-miller-valasek`, 2016. [Accessed on: 15-December-2018].

[15] Charlie Miller and Chris Valasek. Black Hat USA 2016: That Jeep Was Hacked Again. Kaspersky, `https://www.kaspersky.com/blog/jeep-hacked-again/12752`, 2016. [Accessed on: 15-December-2018].

[16] Tao Zhang, Helder Antunes, and Siddhartha Aggarwal. Defending Connected Vehicles Against Malware: Challenges and a Solution Framework. *IEEE Internet of Things journal*, 1(1):10–21, 2014.

[17] Marko Wolf, André Weimerskirch, and Christof Paar. Security in Automotive Bus Systems. In *Workshop on Embedded Security in Cars*, 2004.

[18] Pal-Stefan Murvay and Bogdan Groza. Source Identification Using Signal Characteristics in Controller Area Networks. *IEEE Signal Processing Letters*, 21(4):395–399, 2014.

[19] Priyanka Sharma and Dietmar PF Möller. Protecting ECUs and Vehicles Internal Networks. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pages 0465–0470. IEEE, 2018.

[20] Hyun Min Song, Ha Rang Kim, and Huy Kang Kim. Intrusion Detection System Based on the Analysis of Time Intervals of CAN Messages for In-vehicle Network. In *2016 international conference on information networking (ICOIN)*, pages 63–68. IEEE, 2016.

[21] Manne Engelke and Jesper Ivarsson. Intrusion Detection Systems in Trucks - Evaluation of Intrusion Detection Systems in Dynamic Automotive Environments. Master's thesis, Chalmers University of Technology, 2018.

[22] William Stallings and Lawrie Brown. *Computer Security: Principles and Practice.* Hoboken, New Jersey: Pearson Education, Inc., [2018], 2018.

[23] S.680 - SPY Car Act of 2017. `https://www.congress.gov/bill/115th-congress/senate-bill/680`. [Accessed on: 13-February-2019].

[24] H.R.3388 - SELF DRIVE Act. `https://www.congress.gov/bill/115th-congress/house-bill/3388`. [Accessed on: 13-February-2019].

[25] Wilfried Voss. *A Comprehensible Guide to Controller Area Network.* Greenfield, Mass.: Copperhill Technologies Corporation, 2008.

[26] Physical Layer, 500 Kbps (J1939/14 Ground Vehicle Standard). `https://saemobilus.sae.org/content/j1939/14_201110`, October 2011. [Accessed on: 18-February-2019].

[27] Wilfried Voss. *A Comprehensible Guide to J1939.* Greenfield, Mass.: Copperhill Technologies Corporation, 2008.

[28] Common Criteria for Information Technology Security Evaluation. ISO/IEC 15408. Standard, International Organization for Standardization, 2009.

[29] Common Methodology for Information Technology Security Evaluation. ISO/IEC 18045. Standard, International Organization for Standardization, 2008.

[30] Catakoglu Onur, Balduzzi Marco, and Balzarotti Davide. Automatic Extraction of Indicators of Compromise for Web Applications. In *Proceedings of the 25th International Conference on World Wide Web*, pages 333–343. International World Wide Web Conferences Steering Committee, 2016.

[31] Michael Müter and Naim Asaj. Entropy-based Anomaly Detection for In-vehicle Networks. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 1110–1115. IEEE, 2011.

[32] Christian Sandberg Atul Yadav. HoliSec Reference Architecture-Holistic Approach to Improve Data Security. Technical report, The HoliSec Consortium, 2018.

[33] Holistic Approach to Improve Data Security of Vehicles. `https://www.ri.se/en/what-we-do/projects/holistic-approach-improve-data-security-vehicles`, 2015. [Accessed on: 6-May-2019].

[34] Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. `http://www.rfc-editor.org/rfc/rfc6335.txt`, August 2011. [Accessed on: 16-May-2019].

[35] Prajwol Kumar Nakarmi, Karl Norrman. Detecting false base stations in mobile networks. https://www.ericsson.com/en/blog/2018/6/detecting-false-base-stations-in-mobile-networks, June 2018. [Accessed on: 16-May-2019].

[36] Michael Müter, André Groll, and Felix C Freiling. A Structured Approach to Anomaly Detection for In-Vehicle Networks. In *2010 Sixth International Conference on Information Assurance and Security*, pages 92–98. IEEE, 2010.

[37] Charlie Miller, Chris Valasek. Advanced CAN Injection Techniques for Vehicle Networks. https://youtu.be/4wgEmNlu20c?t=1918, November 2016. [Accessed on: 23-May-2019].

[38] D. K. Nilsson and U. E. Larson. Secure Firmware Updates over the Air in Intelligent Vehicles. In *ICC Workshops - 2008 IEEE International Conference on Communications Workshops*, pages 380–384, May 2008.

[39] H. A. Odat and S. Ganesan. Firmware Over The Air for Automotive, FOTAMOTIVE. In *IEEE International Conference on Electro/Information Technology*, pages 130–139, June 2014.

[40] Mathias Johanson, Pål Dahle, and A Soderberg. Remote Vehicle Diagnostics over the Internet Using the DoIP Protocol. In *The Sixth International Conference on Systems and Networks Communications*, 2011.

[41] Van den Herrewegen, Jan and Garcia, Flavio D. Beneath the Bonnet: A Breakdown of Diagnostic Security. In *European Symposium on Research in Computer Security*, pages 305–324. Springer, 2018.

[42] Kyong-Tak Cho, Yuseung Kim, and Kang G Shin. Who Killed My Parked Car? *arXiv preprint arXiv:1801.07741*, 2018.

[43] Yilu Zhang, Gary W Gantt, Mark Rychlinski, Ryan Edwards, John Correia, and Calvin Wolf. Vehicle Design Validation via Remote Vehicle Diagnosis: A feasibility study on battery management system. In *2008 International Conference on Prognostics and Health Management*, pages 1–6. IEEE, 2008.

[44] Luis Serrano, Donghyun Kim, Richard B Langley, Kenji Itani, and Mami Ueno. A GPS Velocity Sensor: How Accurate Can It Be?–a First Look. In *ION NTM*, volume 2004, pages 875–885, 2004.

[45] Madjid Tavana Fatos Xhafa, Srikanta Patnaik. *Advances in Intelligent, Interactive Systems and Applications.* Springer, 2019.

[46] Common Criteria Recognition Arrangement Members. Common Criteria for Information Technology Security Evaluation. Technical report, Common Criteria Interpretation Management Board, 2017.

[47] Marko Wolf and Michael Scheibel. A systematic approach to a qualified security risk analysis for vehicular it systems. *Automotive-Safety & Security 2012*, 2012.

[48] Mafijul Md Islam, Aljoscha Lautenbach, Christian Sandberg, and Tomas Olovsson. A Risk Assessment Framework for Automotive Embedded Systems. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, pages 3–14. ACM, 2016.