Cybersecurity Verification and Validation Testing in Automotive

Damjan Ekert  
(ISCN GesmbH, Graz, Austria, https://orcid.org/0000-0001-9301-242X, dekert@iscn.com)

Jürgen Dobaj  
(TU Graz, Graz, Austria, https://orcid.org/0000-0001-6460-8080, juergen.dobaj@tugraz.at)

Alen Salamun  
(Real Security, Maribor, Slovenia, alen.salamun@real-sec.com)

Abstract: The new generations of cars have a number of ECUs (Electronic Control Units) which are connected to a central gateway and need to pass cybersecurity integration tests to fulfill the homologation requirements of cars. Cars usually have a gateway server (few have additional domain servers) with Linux and a large number of ECUs which are real time control of actuators (ESP, EPS, ABS, etc. – usually they are multicore embedded controllers) connected by a real time automotive specific bus (CAN-FD) to the domain controller or gateway server. The norms (SAE J3061, ISO 21434) require cybersecurity related verification and validation. For the verification car manufacturers use a network test suite which runs > 2000 test cases and which have to be passed for homologation. These norms have impact on the way how car communication infrastructure is tested, and which cybersecurity attack patterns are checked before a road release of an ECU/car. This paper describes typical verification and validation approaches in modern vehicles and how such test cases are derived and developed.

Keywords: Automotive Cybersecurity, Verification, Validation, Best Practice Design Patterns

Categories: D.2.1, D.2.2, D.2.5, D.4.6

DOI: 10.3897/jucs.71833

1 Introduction

In Feb. 2021 VDA (German Automotive Association) AK 13 [ASPICE 2020] published the Automotive SPICE for cybersecurity assessment model (Figure 1). This will be used for cybersecurity homologation assessments which are mandatory for car makers from July 2022.
The model includes new processes:

- **MAN.7 Cybersecurity Risk Management** (Management, performing and tracking a **TARA** – Cybersecurity Threat Analysis and Risk Analysis, and deriving cybersecurity goals)
- **SEC.1 Cybersecurity Requirements Elicitation** (Requirements derived from a **TARA**)
- **SEC.2 Cybersecurity Implementation** (Designing counter measures against the risks and threats and to achieve the cybersecurity goals)
- **SEC.3 Risk Treatment Verification** (verification against the cybersecurity requirements)
- **SEC.4 Risk Treatment Validation** (validation against the cybersecurity goals)

And extended 2 existing processes:

- **ACQ.2 Supplier Request and Selection** (including cybersecurity in the tender)
- **ACQ.4 Supplier Monitoring** (monitoring the fulfilment of cybersecurity goals work products, requirements)

This paper addresses how the test concepts in cybersecure vehicle architectures are implemented in the SEC.3 and SEC.4 processes.
2 Cybersecurity Testing Related Work Products

In a working party SOQRATES a group of leading automotive suppliers, universities and training bodies and the EU Blueprint project DRIVES [Messnarz 2020] for automotive developed a concept of what typical work products are expected. Figure 2 illustrates the typical work products in listings for 04 to 06.

Figure 2: Cybersecurity related work products 04 – 06 for cybersecurity testing

SEC.3 Risk Treatment Verification: SW Unit Verification (see Figure 2 above)

04 MISRA and CERT

• Cybersecurity coding rules MISRA C:2012 Amendment 1 from 2016, and ISO/IEC TS 17961:2013 C-Secure

04 MITRE guidance

• Cybersecurity patterns (list of attacks and recommended actions) https://attack.mitre.org
• 04 OWASP
• The Open Web Application Security Project (owasp.org) with good and bad practices for programming in web based secure environments

04 Code Inspection

• Review checklists are extended to include cybersecurity relevant aspects. Typically coding rules are checked automatically. However, cybersecurity relevant SW parts need to be marked, need to be independent from the normal code (freedom from interference) applying an extended cybersecurity related checklist and rule set
SEC.3 Risk Treatment Verification: SW Integration / System Integration / Network Test (see Figure 2 above)

05 NTS Network-Test-Suite
- This is a test suite which is connected by a bus (LIN, CAN, CAN-FD, Ethernet) to the ECU (Electronic Control Unit) and has a programmable interface to write test cases (e.g. a replay attack). automatically performs tests and reports cybersecurity coverage.
- Advanced customers who have already an advanced cybersecurity architecture in place usually require to use a specific test suite and run given customer diagnostic tests on the ECU.

05 Types of Network Test Cases
- Test case types are usually structured by cybersecurity objectives and diagnostic chapters
  - e.g. tests for authentication
    - e.g. tests for secure diagnostics and flashing
    - e.g. test for secure boot
    - e.g. tests for replay attacks
    - e.g. tests for UDS (Unified Diagnose Services) diagnostic functions and trying to access secure parts (normally UDS $27) by non-authenticated diagnostic protocol functions
    - e.g. denial of service attacks
    - e.g. behaviour in case messages get lost, come too often in incorrect time frame etc.

05 Attack Types Tested
- All attack types (usually the MITRE catalogue is checked and those parts which are applicable the product being developed are selected) must be covered.

02 Typical OEM Test Cooperation
- Advanced car manufacturers such as VW, Daimler, BMW, etc. provide requirements for the test tool and test procedure set up: e.g. NTS Network Test Suite for Daimler AG, FAT Tool Suite for BMW, CANoe with CAPL programmed test cases in a combination with a HIL platform (HIL – Hardware in the Loop) in case of VW, etc.

SEC.4 Risk Treatment Validation: Software & System Test (see Figure 2 above)

06 Test Bench for Cybersecurity
- Cars in production at the end of line test receive an IP address and all certificates and keys configured. In a test bench these key updates are tested.
- 06 Penetration Test / Test Team Cooperation
  - Penetration testing is done an external cybersecurity teste team. This team does not receive all internal designs but the cybersecurity goals and technical data about the ECU.

06 Vehicle Integration Testing
- Vehicle integration requires now e.g. that the test driver flashes new keys in the vehicle set up.
3 Cybersecurity Requirements for Verification

The cybersecurity norms require to perform a TARA (Threat Analysis and Risk Analysis). The TARA attributes are described in the ISO 21434 norm [SAE 2016][Schmittner 2019][Schmittner 2019 2][ISO 21434 2020][VDA 2021] and a TARA delivers an impact level, a threat level and a security level (combination of impact and threat level by a risk table). [ISO 21434 2020][Macher 2016][Macher 2017][Macher 2017 2] Also, each line in the TARA includes a cybersecurity goals, if the security level shows a security rating (see Figure 4 example extract from a TARA).

Before a TARA is performed a cybersecurity item analysis or threat model is drawn which shows the system, all critical interfaces and critical data and groups the assets which can be attacked [Dobaj 2018][Dobaj 2019][Macher 2020].

See the example in Figure 3 which describes a 6 phase e-motor EPS (Electric Power Steering). One critical interface is e.g. the steering angle request which can lead to unwanted steering of the vehicle [Macher 2019][Macher 2019 2][Macher 2019 3][Messnarz 2016][Messnarz 2016 2][Messnarz 2017][Armengaud 2019][Messnarz 2019 2][Veledar 2019].

Figure 4 shows example lines and selected attributes of the related TARA where the ADAS steering angle request as a critical signal is rated with a threat level medium, and because (not visible) the impact level was critical (since sudden steering leads almost always to an accident), the security level is high.

The assigned security goal in Figure 4 e.g. is “Prevent unwanted steering due to unauthorized commands and assure secure logging of received commands over a period of 800 ms.

The cybersecurity goal is then broken down to system requirements (authorize and authenticate commands) and cybersecurity critical SW function and cybersecurity critical software data requirements. For the critical function requirements related critical data are referenced, the SW states in which the function is active (white listing) is defined, and security objectives are assigned (see Figure 5).
Ekert D., Dobaj J., Salamun A.: Cybersecurity Verification and Validation Testing...

Figure 3: Cybersecurity item analysis – Critical assets and interfaces

Figure 4: Cybersecurity goals – results from a TARA
For each cybersecurity objective corresponding counter measures are programmed and need to be tested.

<table>
<thead>
<tr>
<th>Threat</th>
<th>Property</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoofing</td>
<td>Authentication</td>
<td>Impersonating something or someone else.</td>
<td>Pretending to be a specific device on the vehicle bus, sending out signals and commands.</td>
</tr>
<tr>
<td>Tampering</td>
<td>Integrity</td>
<td>Modifying data or code</td>
<td>Modifying configuration files or firmware storage devices, or modify messages as they traverse the NW.</td>
</tr>
<tr>
<td>Repudiation</td>
<td>Non-repudiation</td>
<td>Claiming to have not performed an action.</td>
<td>An attacker succeeded to modify some data within a storage or a message, and can pretend to have done it.</td>
</tr>
<tr>
<td>Information Disclosure</td>
<td>Confidentiality</td>
<td>Exposing information to someone not authorized to see it.</td>
<td>Reading key material from storage, an application, messages in transit.</td>
</tr>
<tr>
<td>Denial of Service</td>
<td>Availability</td>
<td>Deny or degrade service to users</td>
<td>Crashing/deactivating a device on the bus, sending messages to absorbing CPU resources, flooding the bus, ...</td>
</tr>
<tr>
<td>Elevation of Privilege</td>
<td>Authorization</td>
<td>Gain capabilities without proper authorization</td>
<td>Allowing a remote user to execute commands on the vehicle internet gateway (i.e., the OTA gateway) to send messages on the vehicle bus.</td>
</tr>
</tbody>
</table>

Continuing with the example from steering a MAC will be used for only accepting authenticated and authorized messages:

The MAC (Message Authentication Codes) shall provide the following security properties/objectives:
- freshness (Sec. Objective)
- authentication (Sec. Objective)
- information/payload: steering angle request
- integrity (Sec. Objective)

Algorithmic Details we may assign to the SW Level Analysis:
- freshness: a new random number is added to each message
- authentication: via shared secret
- information/payload: steering angle request
- integrity: via HASH plus CRC for error correction from safety
- CAN-message = [h1][steering angle request][CRC]
- p1 = [shared-secret | increase-of-random-number | steering angle request]
- h1 = HASH[shared-secret | HASH[p1] ]
4 Integration Testing Verification Strategy - Network Test Suites Approach

Figure 7 illustrates a typical general test framework [Akka 2021][Vector 2021] applied and Figure 7 shows the specific test environment which typically is used, for instance, in Daimler projects.

![Figure 7: Typical setup of network test suites](image)

A tool affordable for research teams, and small companies is PCAN where you can program test cases in C/C++ and even in Visual Basic. There is also a whole set functions to import DBC files, sniff the bus, and change, adapt and replay messages.

In vehicles a CAN bus is based on a norm J1939 (https://wiki2.org/en/SAE_J1939) in which the message format is standardized (see Figure 9). Research is done to create more secure protocols based on Ethernet [Dobaj 2020].

A CAN FD [ISO 11898] is structured like a CAN and allows Flexible Data (FD) rates. To add to a CAN a cybersecurity (which was not meant at the beginning) a cybersecurity related message bundle is created which contains both, the message and the cybersecurity related message part to authenticate, authorize, check the command/message (see Figure 8 to Figure 10).

It is important not to confuse security with the built in CRC checksum algorithms (see Figure 8) which support a 15-bit, 17-bit, or 21-bit CRC. This CRC is used for an E2E (End to End) validation of messages, that messages have not been altered in general.

The added SecOC bundle is the protection part against the attackers (see h1 in Figure 8).
Ekert D., Dobaj J., Salamun A.: Cybersecurity Verification and Validation Testing ...

For the design of a vehicle (a vehicle contains many ECUs Electronic Control Units connected by a bus) the CAN message catalogue is specified according to a norm (J1939) and imported by the ECUs. ECU software which is developed for cybersecurity has usually a standard service architecture supported by Autosar >= 4.3 (see Figure 10) which offers services to manage secure communication (SEcOC component), connects to a hardware security module (vHSM and HSM), and allows functional programs to use encryption, decryption, signing, authentication, hash algorithms, etc. The decrypted data are made available via the RTE (Run Time Environment).

**Figure 8: CAN FD Structure and Extension for Cybersecurity**

For the design of a vehicle (a vehicle contains many ECUs Electronic Control Units connected by a bus) the CAN message catalogue is specified according to a norm (J1939) and imported by the ECUs. ECU software which is developed for cybersecurity has usually a standard service architecture supported by Autosar >= 4.3 (see Figure 10) which offers services to manage secure communication (SEcOC component), connects to a hardware security module (vHSM and HSM), and allows functional programs to use encryption, decryption, signing, authentication, hash algorithms, etc. The decrypted data are made available via the RTE (Run Time Environment).

**Figure 9: Message Structure without Security Bundle**

Figures 9 and 10 show a typical outcome in the RTE, where you can read the requested motor torque and also you can read by a flag if the authentication was ok.
In testing Automotive uses an integrated test equipment (Figures 7 and 8) which allow to change, replay, etc. messages and observe the outcome and read by a protocol the values from the RTE.

5 Test Case Design for Verification

Test cases in Automotive need to be linked to the corresponding requirements. This traceability of requirements to test cases and test results is a must in Automotive to prove that features are complete when homologating a vehicle. Different norms check this traceability such as Automotive SPICE [ASPICE 2017 ASPICE 2017 2] and Automotive SPICE extension for Cybersecurity [ASPICE 2021], Functional Safety, and cybersecurity norms like ISO 21434 for proof of coverage of a cybersecurity case. Assessments are made to check this coverage and if the development process supports this [Ekert 2020][Höhn 2015][Messnarz 2007][Messnarz 2009][Messnarz 2012][Messnarz 2019][Schlager 2018][Wegner 2020].

Since the tool setup is supported by a test framework with scripting the test cases usually are designed in a database, linked to requirements and scripts write test results logs which can be imported back to the database to provide a test status.

In SOQRATES best practice attributes for describing such test cases are discussed and in Figure 11 below you find such an example specification using attributes like:

- Text case ID
- Related SW Security Requirement(s)
- Test Case Title
- Test Case Description / Test Steps
- Preconditions, Expected Results
- Test Case Design method used
- Test Level
- Test (Tool) Environment
- Cybersecurity Test level
<table>
<thead>
<tr>
<th>Test ID</th>
<th>Reference for Trace</th>
<th>Status</th>
<th>Test Case Title</th>
<th>Test Descriptions/Test Status</th>
<th>Preconditions</th>
<th>Departed Results</th>
<th>Test Case Design Method (e.g. Equivalence Class, Layered Diagrams)</th>
<th>Fault Inversion, Negative Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC_01</td>
<td>Sw_SEC_F_002</td>
<td>New</td>
<td>Stealing request with wrong hash key</td>
<td>Test case execution with ID 001 in stage 5, 6, 7, and 8. Test successful.</td>
<td>Step 1: TC, MOTOR: CMD, new = 40K, EDP, old = 30.</td>
<td>Step 2: TC, MOTOR: CMD, new = 40K, EDP, old = 30.</td>
<td>Step 3: TC, MOTOR: CMD, new = 50K, EDP, old = 30.</td>
<td>Use the valid command STEER, MOTOR, TORQUE, and measure the hash key. TC, MOTOR: CMD, hash = 40K, EDP, old = 30K, Torque = 30K.</td>
</tr>
<tr>
<td>TC_02</td>
<td>Sw_SEC_F_002</td>
<td>New</td>
<td>Stealing request with wrong hash key</td>
<td>Test case execution with ID 002 in stage 5, 6, 7, and 8. Test successful.</td>
<td>Step 1: TC, MOTOR: CMD, new = 45K, EDP, old = 30.</td>
<td>Step 2: TC, MOTOR: CMD, new = 45K, EDP, old = 30.</td>
<td>Step 3: TC, MOTOR: CMD, new = 50K, EDP, old = 30.</td>
<td>Use the valid command STEER, MOTOR, TORQUE, and measure the hash key. TC, MOTOR: CMD, hash = 45K, EDP, old = 30K, Torque = 30K.</td>
</tr>
</tbody>
</table>

Figure 11: Cybersecurity verification test case example
6 Test Case Approach in Case of Validation

While the cybersecurity verification gives the test team a white box view with a number of cybersecurity related function and data requirements, the validation is based on a black box view. The testers know the functions, the technical description and the cybersecurity goals, but they do not receive in e.g. penetration testing the details. This is used to simulate a real case attacker from outside.

Also, attacker teams for penetration test have a structured test strategy which is called “Creating Adversary Emulation Plans based on attack patterns derived from MITRE” (Figure 12) [MITRE 2021][MITRE 2021 2].

Using the test specification, the editors and scripts in the tools framework are used to e.g. resend the message with wrong message counter (which e.g. would happen in a replay attack).

![Example attack flow](image)

**Figure 12: Cybersecurity attack flow analysis**

For the preparation of such plans the MITRE Also, attacker teams for penetration test have a structured test strategy which is called “Creating Adversary Emulation Plans based on attack patterns derived from MITRE” (Figure 12).

Also, attacker teams for penetration test have a structured test strategy which is called “Creating Adversary Emulation Plans based on attack patterns derived from MITRE” (Figure 12).

In Mitre.org you can select the attack navigator functionality, create a new layer model for you, and start planning. This might result in a strategy described in Figure 13.
MITRE is based on a collection of attack patterns observed in ICT industry, and thus the vehicle testers need to adapt the proposed ICT strategy to a vehicle based strategy. This results in an attack strategy as described in Figure 14.

This way pen Testing structured approach based on the assumed objectives of an adversary pan that was derived from MITRE and adapted for automotive.

The penetration test cases are then grouped by these attack flows / scenarios

7 Conclusions
Modern cars have a gateway server with an IP address and are connected to the network and the server / gateway can be attacked [Macher 2019] [Messnarz 2020]. Cars will in future expand into e-city, e-environment modes building a connected society [Feuer 2002] [Messnarz 2020].
Cybersecurity norms like ISO 21434 and legal regulations for cars in UNECE law require the coverage of cybersecurity norms and requirements and to prove that by cybersecurity verification and validation.

Automotive SPICE for cybersecurity has been recently published in Feb. 2021 and has processes included for cybersecurity verification and cybersecurity validation.

This paper described how test strategies in cybersecure vehicles are implemented and how this maps onto a cybersecure vehicle architecture.

Acknowledgements

We are grateful to the guest editors Dr Richard Messnarz from ISCN GesmbH and Dr Georg Macher from TU Graz who are experts of the cybersecurity task force in the German and Austrian working group SOQRATES (soqrates.eurospi.net) and who contributed state of the art experiences in the field.

We are grateful to the European Commission which has funded the BLUEPRINT project DRIVES (2018 – 2021) [Messnarz 2020] [DRIVES 2021] [Stolfa 2020]. In this case the publications reflect the views only of the author(s), and the Commission cannot be held responsible for any use, which may be made of the information contained therein.

We are grateful to the EuroSPI community and conference series (www.eurospi.net) in which experts share experiences since 1994 and cybersecurity and JUCS has been promoted and experts from that community submitted papers to this journal.

We are grateful to the EU Project ECQA Certified Cybersecurity Engineer and Manager – Automotive Sector, Erasmus+ Programme, Grant Agreement No. 2020-1-

CZ01-KA203-078494.

We are grateful to a working party of Automotive suppliers SOQRATES [SOQRATES 2021] (https://soqrates.eurospi.net) who exchange knowledge about such assessment strategies. This includes:

Aschenberger Werner (KTM AG), Bauer-Wukitsevits Robert (msg Plaut), Böhner Martin (Elektrobit), Brasse Michael (HELIA), Bressau Ernst (BBraun), Dallinger Martin (ZF), Dorociak Rafal (HELIA), Dreves Rainer (Continental), Ekert Damjan (ISCN), Forster Martin (ZKW), Geipel Thomas (BOSCH Engineering), Grave Rudolf (Elektrobit), Griessnig Gerhard (AVL), Gruber Andreas (Kreisel Electric), Habel Stephan (Continental), Hällmayer Frank (SF), Haunert Lutz, Karner Christoph (KTM AG), Kinalzyk Dietmar (AVL), König Frank (ZF), Lichtenberger Christoph (MAGNA), Lindermuth Peter (MAGNA), Macher Georg (TU Graz), Maier Bjørn (AVL), Mandic Irenka (MAGNA), Marc Dijaz (LORIT Consultancy), Mayer Ralf (BOSCH Engineering), Mergen Silvana (TDK EPCOS), Messnarz Richard (ISCN), Much Alexander (Elektrobit), Nikolov Borislav (msg Plaut), Oehler Couso Daniel (MAGNA), Riel Andreas (ISCN Group), Rieß Armin (Braun), Santer Christian (AVL), Schlager Christian (MAGNA), Schmittner Christoph (AIT),

Schubert Marion (ZKW), Sechser Bernhard (Process Fellows), Sokic Ivan (Continental), Sporer Harald (Infineon), Stahl Florian (msg Plaut), Wachter Stefan (msg Plaut), Walker Alastair (LORIT Consultancy), Wegner Thomas (ZF), Geyer Dirk (AVL), Dobaj Jürgen (TU Graz), Wagner Hans (msg Group), Aust Detlev (AustCON).
References


Ekert D., Dobaj J., Salamun A.: Cybersecurity Verification and Validation Testing ... 865

Improved Trustworthiness and Weather-Independence of Conditionally Automated Vehicles in
Mixed Traffic Scenarios International Forum on Advanced Microsystems for Automotive
Applications, 2018, 75-89

[ISO 11898] CAN FD, ISO 11898-1 Road vehicles — Controller area network (CAN) — Data
link layer and physical signalling

[ISO 21434 2020] ISO 21434 ISO/SAE 21434 DIS, Road vehicles – Cybersecurity engineering,
DIS version, Feb 2020


2nd Edition Road vehicles Functional Safety”, 2018

Computer and Information Science, vol 1060. Springer, Cham. https://doi.org/10.1007/978-3-
030-28005-5_4

Computer and Information Science, vol 1251. Springer, Cham. https://doi.org/10.1007/978-3-
030-56441-4_9

[Macher et al., 2016] Macher, G.; Sporer, H.; Brenner, E. & Kreiner, C. “Supporting Cyber-
security based on Hardware-Software Interface Definition Systems”, Software and Services

SPICE, Safety and Cybersecurity Integration. In: Tonetta S., Schoitsch E., Bitsch F. (eds)
Science, vol 1251. Springer, Cham

Development in the Automotive Domain, Working Group 17AE-0252/2017-01-1661, SAE
International, June 2017

An integrated view on automotive spice, functional safety and cyber-security. In 2020 SAE
Technical Papers 2020-01-0145, SAE International.

Aspects of Fail-Operational Urban Surround perception (FUSION), International Symposium
on Model-Based Safety and Assessment, 2019, 286-300

Quest for Infrastructures and Engineering Methods Enabling Highly Dynamic, Autonomous
Systems European Conference on Software Process Improvement, 2019, 15-27

21434 Automotive Cybersecurity Standard-In a Nutshell. In 2020 International Conference on


