Security Crash Test – Practical Security Evaluations of Automotive Onboard IT Components

Stephanie Bayer, Thomas Enderle, Dennis Kengo Oka[*)], Marko Wolf

ESCRYPT GmbH
Leopoldstraße 244
80807 München
Germany

[*)] ETAS K.K.
Queens Tower C-17F
2-3-5, Minatomirai, Nishi-ku
Yokohama, 220-6217 Japan

{stephanie.bayer; thomas.enderle; marko.wolf}@escrypt.com
dennis-kengo.oka@etas.com

Abstract: Modern vehicles consist of many interconnected, software-based IT components which are tested very carefully for correct functional behavior to avoid safety problems, e.g. that the brakes suddenly stop working. However, in contrast to safety testing systematic testing against potential security gaps is not yet a common procedure within the automotive domain. This however could eventually enable a malicious entity to be able to attack a safety-critical IT component or even the whole vehicle. Several real-world demonstrations have already shown that this risk is not only academic theory [1].
Facing this challenge, the paper at hand first introduces some potential automotive security attacks and some important automotive security threats. It then explains in more detail how to identify and evaluate potential security threats for automotive IT components based on theoretical security analyses and practical security testing. Lastly, we propose "automotive security evaluation assurance levels" (ASEAL) which define up to four discrete security testing levels.

1 Introduction

Suddenly car drivers all over the world witness spooky behavior of their Internet-enabled car infotainment units over the past few days. Out of the blue their navigation system jumps to another route, the unit calls the service on its own, or the display shows sculls and laughing, white masks. A quick analysis by security experts shows that the reason behind this behavior is a critical security breach of the GSM/LTE interface that enables unauthorized persons to access the software of the infotainment unit. But, how is it possible that this vulnerability had been missed, as the infotainment unit passed numerous tests? The answer is quite simple; even though there were several tests focusing on the functional safety, a systematic security evaluation containing theoretical analysis and practical tests had not been accomplished.

Luckily, this is only a potential scenario (yet) and not a real case but it clearly exposes a critical gap in automotive IT testing, which is not yet covered by existing functional testing procedures that are already well-established and conducted since several decades.

However, in contrast to functional testing, the systematic evaluation of automotive IT components regarding IT security is still in a very early stage. At the same time there is a strong need for security testing to be a part of the engineering procedure, not only due to scenarios like the introductive example, but also as a result of corresponding research activities [2], [3] and increasing demands made by public authorities [4]. Hence, this paper starts with a discussion about advantages and corresponding efforts of systematic security testing for the automotive industry. To do this we take into account recent security threats for automotive IT systems, ranging from odometer manipulation up to remote controlling of the steering system, and explain in detail why good security evaluations could have prevented most of the attacks.

The contribution of this paper is two-fold. First, we give an overview and a short introduction to the different aspects of embedded security evaluations such as theoretical security analyses, practical security testing, and verifiable security verification (cf. Figure **2**) especially regarding automotive onboard IT components.

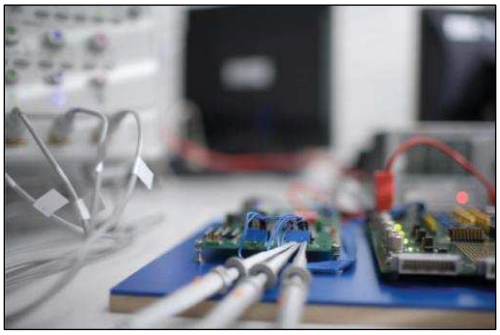

Figure 1: A practical testing setup for testing an embedded device at the authors' lab.

The second contribution of this work is a first proposal for establishing so-called "automotive security evaluation assurance levels" (ASEAL) which define up to four discrete security testing levels that determine which security analyses and tests should be part of a certain ASEAL and how "deeply" these security analyses and tests should be executed (cf. Section Table 1).

2 Automotive Security Threats

This section presents an overview of potential automotive security threats. To understand the type of threats, one must first have a basic understanding of the various automotive functionalities which can be targeted. Obviously there are direct security threats on driving safety such as manipulating the steering wheel or brakes, and indirect security threats such as distracting the driver by triggering odd vehicle behavior. Security researchers have successfully demonstrated that it is already possible to engage or disable the brakes or manipulate the steering wheel [3], [2], [1], [5] by maliciously injecting the relevant messages on the vehicle CAN bus. Indirect safety issues are

possible by injecting CAN messages to, for example, disable wipers or turning off the headlights when it is raining and dark outside.

Another type of security threat is targeting authoritative functionalities in the vehicle. For instance, the odometer logs the traveled distance, and, when selling a used car, it is attractive for an attacker to reduce the odometer value to increase the value of the car. In Germany, according to police investigations, around 2 million cars are subject to odometer manipulation per year with an average loss per vehicle of around 3000 € resulting in total losses of around 6 billion € per year [6]. Furthermore, critical data is stored in the ECUs such as crash data, data for insurances, or warranty indicators. Such data is also very attractive for malicious manipulations. For example, data such as vehicle speed, seat belt status, brake pedal position etc. are typically recorded in the seconds before a crash. A driver who has been involved in an accident could be motivated to change the recorded data to indicate that the brakes were applied when they really were not.

Moreover, since vehicles are becoming ubiquitously interconnected, increasing amounts of private data such as vehicle location, credentials to online services, or mobile payment data are stored in the vehicle as well. Attackers may be interested in stealing such data and misuse them directly or use as a stepping stone to launch further attacks, for instance, attacking an online service by stealing the respective credentials stored in the vehicle. Such private data could be extracted wirelessly by exploiting security vulnerabilities in services provided by communication interfaces such as Bluetooth and Wi-Fi, or through physical access to the OBD port, a USB port, or the ECU itself.

Another type of threat is theft of vehicles or valuable vehicle components such as airbags or head units, e.g. by abusing diagnostics commands to reprogram a new key. This functionality is typically used by workshop dealers when replacing a lost key, but can also be exploited by attackers to program a “thief key” for the vehicle that they are stealing [7]. There are other cases where attackers are able to send control messages to a vehicle to disable the alarm and unlock the doors, resulting in attackers being able to gain physical access to the interior of a vehicle [8].

3 Related Work

There is extensive work in IT automotive security testing conducted mostly by independent security researchers. For example, the Center for Automotive Embedded Systems Security (CAESS) [3], [2] has performed practical security evaluations of vehicles and found a number of issues. They have successfully exploited these vulnerabilities resulting in disabling of engine or brakes as well as in gaining remote access to internal systems. Both security testing of external interfaces to remotely access the vehicle as well as security testing of the in-vehicle network was performed. Other researchers [1] have shown practical attacks based on non-diagnostic CAN messages resulting in killing the engine, locking the brakes, or jerking the steering wheel while driving.

Lately, there have been various requests that urge OEMs to consider security and to perform more rigorous security testing. For example, Senator Edward Markey has sent a letter to 20 automotive manufactures inquiring about their view of and commitment to security [4]. Moreover, an independent group called I am the cavalry [9] has issued a 5 point security guide for automotive manufacturers in August 2014 urging them to follow this guide to improve automotive security. The five points are: Safety by design, Third party collaboration, Evidence capture, Security updates, and Segmentation and isolation.

Moreover, there exists an annual event [10] where OEMs provide vehicles for automotive security testing by independent researchers, engineers and students. The key idea is that the participating members at the event practically test the security of the vehicles by trying to find vulnerabilities. The OEMs typically have engineers participating in the event to understand what the specific vulnerabilities are as a short-term goal and to learn to consider security in the design (to get the mindset of an attacker when designing the solution) as a long-term goal.

Other industries also make use of security testing. For example, for ICS (industrial control systems), there is an EDSA (embedded device security assessment) certification [11] that includes practical security testing in terms of fuzz testing for the communication robustness testing component. For the banking industry, there is EMVCo certification [12] that includes a Security Evaluation process where general security performance characteristics and the suitability of use for smart card related products and IC (integrated circuits) chip-based tokens are evaluated. The purpose of the EMVCo evaluation is to assess whether the security features provided by the chip product are appropriately implemented. Furthermore, practical testing including penetration testing examines the interaction between the chip, operating system, and application to evaluate whether sensitive and secret information, as well as payment assets are adequately protected by the final chip product.

For the computer security certification, there is an international standard (ISO/IEC 15408 [13]) called Common Criteria for Information Technology Security Evaluation (CC). To achieve a particular Evaluation Assurance Level (EAL), the computer system must meet specific assurance requirements. These requirements typically involve design documentation, design analysis, functional testing and penetration testing.

In the US, requirements for government security are regulated by Federal Information Processing Standards (FIPS) [14]. The purpose for FIPS validation is that it assures users that a given technology has passed rigorous testing under either the Cryptographic Algorithm Validation Program (CAVP) or Cryptographic Module Validation Program (CMVP) by an accredited third-party lab and can be used to secure sensitive information. There exist several FIPSs, for example FIPS 140-2 [15] provides security requirements for cryptographic modules.

4 Embedded Security Evaluation

As discussed in the Section 2, modern vehicles can be open to various security risks. By applying in-depth security evaluations for an automotive IT system, for instance an

ECU, potential security weaknesses can be identified and countered before an attacker can exploit this weakness in the field and cause real financial or even safety damages. The earlier such a security evaluation is done within the developer cycle the less costly and time-consuming it is as well as security weaknesses can be found early and be closed effectively.

Automotive security evaluations can be done in theory as well as in practice. Theoretical security evaluation can (and should) be done during virtually all steps of the automotive development cycle, ideally already from the very beginning, when only a description of the vehicular IT system is available. Subsequent security evaluations for the next product development iterations can then be done very efficiently based on the results of the previous evaluation. In fact, the need for theoretical and practical security evaluations does not even end with series production of the corresponding IT system. Even in the field the IT component might need a security re-evaluation (and eventually also new countermeasures) due to ongoing development of new attacks or new results from security research. Practical security testing, of course, can only be conducted on an implementation of the target system, for instance with a first prototype. It is important to note that security evaluation can support but cannot replace mandatory security protection measures such as security by design or security engineering. In the following section we will explain the different approaches of security evaluation in embedded systems, see Figure **2**. For each of the three main categories we will shortly address the different sub-categories and explain the benefit of each method.

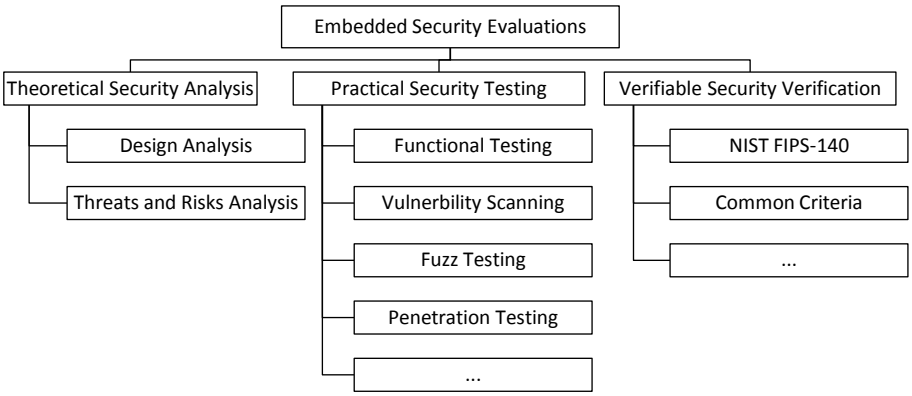

Figure 2: Overview of Embedded Security Evaluation Categories.

4.1 Theoretical Automotive Security Analyses

Theoretical security analyses are becoming gradually more common in the automotive context [16] and are applied to identify and understand the security weaknesses of an automotive IT system based on a paper-based evaluation of the corresponding system specifications and documentations. Depending on the level of scrutiny and the documents available, we differentiate between a more high-level design analysis and a more in-depth threat and risk analysis.

To conduct a **Design Analysis** of an automotive system only a theoretical description of the system is needed. Depending on the level of detail of these descriptions, e.g. high-level protocol descriptions up to explicit specifications, the depth and accuracy of the analysis varies. What is the goal of such a design analysis? First, the analysis can identify systematic flaws in the system even in an early state in the development since high level descriptions can be adequate for a design analysis. Secondly, the results can establish trust in the soundness of the system's architecture. To achieve these goals, the documents are inspected for potential attack points, e.g. weak cryptographic algorithms or possible attacks due to bad interaction of different standard protocols.

To categorize the identified vulnerabilities further and to detect the important flaws of the system that needs to be fixed, a **Threat and Risk Analysis** can be applied to the system. What are the important steps of such a threat analysis? Starting from the available documents, the system is analyzed and possible attacks identified; this step is similar to the design analysis. Additionally the difficulty of the corresponding attack procedure is rated for each of the attacks identified. The rating takes amongst others the required time, the needed expertise of the attacker, the equipment, and the needed access level into account. Furthermore, the potential damage of a successful attack is estimated in terms of safety, operational, and financial damages for the customer. Both values, the attack difficulty and the potential attack damage, result in an overall risk for a certain attack. Security vulnerabilities that result in attacks with a high risk are then critical candidates that should be fixed first.

Nonetheless, theoretical security analyses cannot find any implementation flaws or deviations of the implementation from the specification. Nor can a security analysis find vulnerabilities that are part of insufficiently documented specifications or flaws hidden in supplied components from third parties. To guard the system against such implementation issues, secure software development measures should be applied to the whole vehicular development process [17], [18]. But especially practical security testing, as described in the following section, can be used to identify possible vulnerabilities and cover the gap.

4.2 Practical Automotive Security Testing

Practical security testing can find implementation errors that could be exploited by an outside attacker, but also unspecified functionality and discrepancies to the specifications. Therefore, a thorough practical security test helps to establish trust in the soundness of the implementation. Furthermore, practical security tests help to estimate the actual difficulty of an attack against the target system. In general, practical security testing consists of at least four different steps (Figure 2) as described in the following paragraph.

In the first step, **functional security testing,** tests all security-related functions inside the test system for correct behavior and robustness. This step is similar to general functional testing but with focus on security functionality. A careful execution of this test can find implementation errors, discrepancies to the specification, and especially unspecified functionality that all might result in a potential security weakness. The next step, **vulnerability scanning,** tests the system for already known common security

vulnerabilities, for instance, known security exploits or (security) configurations with known weaknesses. **Fuzzing** goes even further and tries to find new vulnerabilities of an implementation by sending systematically malformed input to the target system to check for unknown, potentially security-critical system behavior. To test the security of the whole system, that means software and hardware, highly individual **penetration tests** can be applied in a last step. During a penetration test a "smart human tester" tries to exploit all the vulnerabilities which were found in the earlier steps in a "sophisticated way" based on many years of "hacking experience" with the goal to change the behavior of the target system. All these methods are explained in more detail in the next Section 5.

However, practical security testing, especially fuzzing and penetration testing, cannot give any assertion on completeness. Depending on the time and resources it is possible to miss larger systematic flaws. Hence, practical security testing cannot replace theoretical security analyses and should always be complemented by a theoretical analysis to identify possible attack paths. Additionally, Secure Software Development should be applied to the whole development process to minimize the total attack surface early on.

4.3 Verifiable Automotive Security Certifications

As discussed in Section 3, Security Certification is not a new idea ([11], [12], [13], [14]); although, there is no direct certification standard for the automotive industry, an overall Security Certification Standard which assures certain levels of security for an automotive system would serve extremely useful. A proposal for a verifiable automotive Security Certifications can be found in Chapter 6.

To certify a system, first theoretical and practical analyses of the system are needed to understand the level of security and the risk of the system. The common idea is that for a certain certification level, a system has to pass a set of tests. In other words, a Security Certificate assures that the process of specification, implementation, and evaluation of the system at hand has been conducted at a standard and repeatable manner with a certain level of security. Therefore**, Security Certifications** help to compare the security level of different target systems and to build trust with customers.

5 Practical Automotive Security Testing

Corresponding publications from the last years, e.g. [16] and [19], focus on theoretical security evaluation without detailing the importance of practical security testing. Within this section, we close this gap and explain why the automotive industry can gain significantly especially from practical security testing.

As explained earlier, a good theoretical security evaluation can spot many issues and counter them; however, sometimes even a well-designed automotive system can suffer from substandard implementations, poor configurations, or physical weaknesses [20]. For example, the random seed for security access could be obtained from a hardware register but the register has no entropy at this stage of the booting process, resulting in a constant seed; hence, we stress the importance of practical security testing.

As illustrated in Figure **3**, there are various techniques to conduct a practical security evaluation. Depending on the time and effort one wants to spend, the range of these tests spans from simple interface scans for known vulnerabilities to invasive techniques such as micro-probing to recover secret data from an automotive component. Other methods, such as fuzzing to find unknown vulnerabilities of a component or power analysis to recover cryptographic keys lie between these extremes. A combination of all these approaches is a powerful method to find security vulnerabilities in automotive components that may have been missed in or have not been covered by theoretical security analyses.

For all the approaches the tester needs to have access to the actual software and hardware of the target system, cf. Figure **1**. Furthermore, for functional testing also the specification of the system is needed, and for all other methods supporting software, for instance restbus simulation, may be needed to run the hardware device. Moreover, special testing hardware and software is needed, for example, JTAG-debuggers or special signal generators.

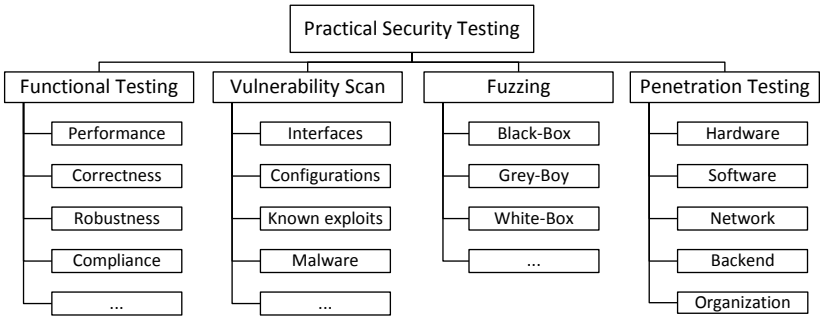

Figure 3 Classification of Practical Security Testing Methods.

5.1 Functional Automotive Security Testing

Functional automotive security testing ensures the general compliance to specifications and standards of the implemented security functionality, for instance, encryption algorithms and authentication protocols, of a vehicular IT system. However, the algorithms are not only tested for correct behavior according to the specification but also for robustness. Furthermore, performance of (often computationally intense) security algorithms is tested to identify potential bottlenecks that might affect the overall security performance. As a result, functional security testing ensures dependable security functionality and that a functional weakness does not create any exploitable security threats.

In many cases, standard implementations, such as OpenSSL [21] are not suitable for use in the automotive domain due to various constraints, and therefore a much wider spectrum of cryptographic and security relevant implementations are in use. Performance or size limitations need to be considered but also safety standards such as MISRA-C [22] must be fulfilled. Furthermore, a wide range of automotive specific security protocols are in use, such as secure flash algorithms or secure communication, secure OBD, theft protection, and upcoming vehicle-to-x (V2X) communication. It is

vital that those security implementations are subject to thorough functional security testing.

Functional security is usually achieved by testing the implementation against official test vectors (if available) or independent implementations. Many cryptographic algorithms could contain specific corner cases that could lead to security vulnerabilities, for example subtle flaws in numeric implementations that trigger only in one out of 4 billion random cases. These corner cases must be tested with specially constructed test vectors and by running lengthy tests. Furthermore, many modern cryptography schemes and security implementations rely on secure random number generators. In order to gain trust into the security of such a random number source, extensive statistical testing is required. Finally, in the highly performance- and cost-sensitive automotive environment, performance testing can help with correct dimensioning of hardware and enable an optimal choice of security algorithms and parameters.

5.2 Automotive Vulnerability Scans

Vulnerability scans are used to examine all relevant applications, source codes, networks, and backend infrastructures of an automotive system for known security weaknesses from a continuously updated database of known automotive security vulnerabilities.

There are numerous different variations of vulnerability scanning. Firstly, the code of the software/firmware running on the system can be scanned, identifying, for example, buffer overflows and heap overflows by using static and dynamic analyses. Depending on the respective analysis tools, this must be done on source level and binary level. Note that it is not necessarily clear that the result is the same in both cases. The compilation process may introduce more security vulnerabilities, for example by removing security checks during the optimization step, or through faulty compilers. Therefore, compiler settings must be examined closely.

Secondly, the system can be scanned for open ports and interfaces, and also for available services running on these interfaces. In automotive systems, this encompasses classical IT interfaces such as IP communication on Ethernet, Wi-Fi, or cellular internet. Scanning these interfaces is especially valuable due to the fact that a whole range of operating systems, network stacks, applications, and libraries are typically re-used, where a large base of vulnerabilities is already known and can be tested for automatically, as done in OpenVAS [23]. Scanning includes reconnaissance port scans, as well as deep scans of specific vulnerabilities. In addition, the automotive environment has special automotive bus systems such as CAN, which have no equivalent in classical IT, but which are highly standardized. This means that automatic scanning tools are well-suited to provide a first overview of vulnerabilities. In this context, scans of diagnostic functionality are notable, as those are likely to contain weakly documented security critical functionality, such as development or debugging functionality.

As a third form of vulnerability scanning, the configuration for the whole system can be analyzed to identify security gaps, e.g. access to critical functions possible without authentication. Automated scans can also test for the presence of different authentication

mechanisms securing the same critical functions. To conclude, vulnerability scanning ensures that a system is secure against known attacks, which can easily be tried out by attackers and therefore are very likely attacks.

5.3 Automotive Fuzzing

Fuzzing is a technique used for a long time to test software and IP networks by exposing the implementation to unexpected, invalid, or random input with the hope that the target will react in an unexpected way, and thereby, to discover new vulnerabilities. The reaction of the target can range from strange output over unspecified behavior up to crashes. Fuzzing as a testing technique for automotive target systems is relatively new; although, modern vehicles have many similarities to common computer networks. In fact, ECUs can be viewed as small computers, running different software, that are connected by different network types such as CAN, FlexRay, or MOST. Hence, it is quite natural to consider the idea to apply fuzz testing also to automotive target systems as part of the security testing process.

In general, fuzzing consists of three different steps: firstly the creation of the input for the target, secondly the delivery of the input to the target and lastly the monitoring of the target system to detect errors in the program flow. Since fuzzing is widely used in the computer world, fuzzing tools such as Peach [24] already exists. Peach has a powerful fuzz generator that can be adapted individually for different protocols such as UDS. The input generated by the fuzz generator is then delivered to the target using the required transport protocol. The target system is monitored to detect possible vulnerabilities. The monitoring process can range from inspection of return values up to the usage of debuggers which observe the internal status of the target device. In the end, all identified unusual behavior has to be analyzed by an expert to detect exploitable vulnerabilities. Examples of such exploitable bugs are insufficient input validation or undocumented functionality, e.g. open debug or configuration interfaces. One famous example for insufficient input validation is the Heartbleed Bug [25] of OpenSSL, which allows reading out critical data since length parameters are not double-checked.

In the automotive context, fuzzing can be applied to diagnostics protocols, such as UDS, or to automotive network protocols, e.g. CAN, FlexRay, MOST or LIN. However, classical fuzzing targets, i.e. IP based networks, play an increasing role in modern vehicles. Hence, automotive security testing also benefits from experiences made in fuzz testing of classical protocols and modern software applications, e.g. cellphone apps.

5.4 Automotive Penetration Testing

Automotive penetration tests are motivated by either IP protection or authoritative functionalities that rely on the integrity of the target system against interests of physically present persons. Examples are theft protection, component protection, odometer manipulations, feature activation, protection from false warranty claims from “tuned” vehicles, or safety functionality. However, in the modern connected world remote attacks begin to be a real threat [26] and this motivates penetration tests of all

communication channels, the backend and relevant organizational processes (e.g. for social engineering attacks).

Typically, penetration tests of a physical device start with general reconnaissance, which includes enumerating interfaces, determining components and their connections on the PCB, gathering specifications available to a hypothetical attacker, and in general, any information that can be helpful in further attacks. Using the information acquired in the first step, further attacks can be planned. A second step may include attacks of local external interfaces such as USB, serial ports, or attacks of the hardware itself. To attack the hardware, usually the tester tries to find overlooked or undocumented debug access interfaces, or gain access to ECU-internal interfaces such as memory buses. More advanced methods require etching open chip packages and accessing the actual silicon chip. In a third step all communication channels to the device, such as the CAN bus, Ethernet, or Wi-Fi, are analyzed and used to are used to attack the target device. Depending on target system and the scope of the penetration test, further attacks against the backend can be conducted. Regarding penetration testing, there are three specific forms: black-box tests, white-box tests, and grey-box tests. In the following we will describe the approaches in more detail.

For black-box testing, the tester is provided with practically no documentation or specifications, except information that could also be acquired by a real world attacker. The advantage of this method is that this results in a very realistic simulation of a real attack. As a disadvantage, the penetration tester must spend a lot of time on basic reverse engineering, and there is a good chance that deeper attack paths are not discovered because the tester did not circumvent easier first-line defense mechanisms, whereas a real attacker may later break such defense mechanisms, by luck, because more information has become public as a result of the state-of-the-art advancing, or because he invested more resources than the tester.

For a white-box test, the tester is provided with full specifications and documentation for the device under test. This means he is able to specifically target weaknesses, and has more resources available, which he did not have to spend on gaining information. Both reasons improve the efficiency of the test. The disadvantage of white-box testing is that the conditions are not nearly as realistic as in black-box tests, giving a less reliable estimation of attack difficulty and likelihood.

Grey-box-tests represent a middle ground between black-box and white-box testing. For a grey-box test, the tester receives partial information, concerning a specific sub-system that is in focus or information that a specific attacker such as an insider could have acquired. A step-wise approach is also possible, where the tester receives more information or access after having shown the basic presence and exploitability of a vulnerability without having to fully develop the attack itself. This optimizes the ratio of test efficiency and realism.

However, penetration attacks are not necessary limited to attacks on the devices hardware, software and networks, but can also include attacks on the organizational implementation like social engineering attacks or weak organizational processes.

6 Automotive Security Assurance Levels

In this chapter we give a first proposal of so-called "automotive security evaluation assurance levels" (ASEAL) which define up to four discrete security testing levels that determine (i) the size of security evaluation scope, that means which security analyses and tests have to be executed for a certain ASEAL and (ii) how "deeply" and thoroughly these security analyses and tests have to be executed. The goal for ASEAL is to make security evaluations comparable and, in consequence, to make it possible to assign standardized levels of minimum security assurance to each automotive onboard IT component. Our ASEAL approach hence combines and extends the rather specific vehicle-to-vehicle communication security evaluation approach using "Trust Assurances Levels" [19] with the rather general ideas from overall security evaluation framework of Common Criteria [13] to form a standardized automotive IT security assurance level.

Concretely, we define – depending on the required ASEAL A, B, C or D – minimum requirements regarding evaluation scope and depth for each theoretical security analysis and each practical security test. Table 1 gives a first (yet incomplete) overview about the idea in general by using exemplary classifications and potential security evaluation assurance levels.

Table 1: Exemplary Automotive Security Evaluation Assurance Levels (ASEAL).

Evaluation Category	Type of Automotive Security Evaluation		Scope and Depth of Automotive Security Evaluation for each ASEAL			
			ASEAL A	ASEAL B	ASEAL C	ASEAL D
Theoretical Security Analyses	TRA: Security Threats and Risks Analysis		TRA1	TRA2	TRA3	TRA4
	SDA: Security Design Analysis		SDA1	SDA2	SDA3	SDA4
	DEV: Security Development Analysis		--	--	DEV1	DEV2
	DEP: Security Deployment & Processes Analysis		--	DEP1	DEP2	DEP3
Practical Security Testing	FST: Functional Security Testing		FST1	FST2	FST3	FST4
	VUL: Vulnerability Scanning		--	VUL1	VUL2	VUL3
	SYF: Systematic Fuzzing			SYF1	SYF2	SYF3
	Penetration Testing	LPA: Logical Penetration Attacks	--	--	LPA1	LPA2
		IPA: Invasive Penetration Attacks	--	--	IPA1	IPA2
		ORA: Organizational Pen. Attacks	--	--	ORA1	ORA2

In the following we present four exemplary definitions (i.e. two theoretical analyses and two practical security tests) for the proposed security evaluation types including some first proposals for the ASEAL-depending evaluation scope and depth.

Table 2: Exemplary ASEAL definitions for automotive security threats & risks analyses

Type	TRA: Security Threats and Risks Evaluation	
Description	Analyzing potential automotive attackers (i.e. owner, garage, competitor, third party), attack scenarios (e.g. undermined business models, parameter manipulation, IP theft, sabotage), and attack paths (e.g. telematics interface, NFC, OBD, Internet).	
For ASEAL A	TRA1	Informal analyses with standard automotive attacker model and attack paths.
For ASEAL B	TRA2	TRA1 + complete attack tree including weighted attack paths for all known automotive security attacks.

For ASEAL C	TRA3	TRA1/2 + methodically tested and verified (cf. CC EAL 4 [13]) including thorough "Darknet" investigations and research for weighting of potentially yet unknown attack paths.
For ASEAL D	TRA4	TRA1/2/3 + semi-formal tested and verified (cf. CC EAL 5/6 [13]).

Table 3: Exemplary ASEAL definitions for security deployment & process analyses

Type	DEP: Security Deployment & Process Evaluation	
Description	Analyzing the security of the component integration, deployment and all other security-related processes at the backend (e.g. key creation/distribution, web interface, access authorizations, and backend security parameters), automotive component (e.g. key injection, access control, initialization, personalization), and communications (e.g. key length, algorithm, key exchange) during production, operation, and phase-out.	
ASEAL A	--	No evaluation required.
ASEAL B	DEP1	Analyses initial setup and initial security configuration.
ASEAL C	DEP2	DEP1 + production, maintenance, and change deployment & processes.
ASEAL D	DEP3	DEP1/2 + phase-out and deactivations processes.

Table 4: Exemplary ASEAL definitions for systematic automotive fuzzing tests

Type	SYF: Systematic Automotive Fuzzing	
Description	Analyzing the security of the target by systematic fuzzing of software communication stacks (e.g. CAN stack, Ethernet stack) and external interfaces (e.g. USB)	
ASEAL A	--	No fuzzing tests required.
ASEAL B	SYF1	Fuzzing of all standard communication protocols (e.g. UDS, TCP/IP) and external interfaces (e.g. Wi-Fi, Bluetooth)
ASEAL C	SYF1	SYF1 + fuzzing of non-standard communication protocols (e.g. special RS232 protocols, proprietary CAN protocols)
ASEAL D	SYF2	SYF2 + extended fuzzing method (i.e. evolutionary fuzzing) for all communication protocols and interfaces

Table 5: Exemplary ASEAL definitions for physical automotive attack tests

Type	IPA: Invasive Penetration Attacks	
Description	Analyzing the attack probability and costs for all kind of attacks (including insider, offline, side-channel, or fault injection attacks) which are very critical for most embedded devices working within a "hostile environment" where attackers have full physical control about the device itself and its environment (e.g. power supply, input signals, temperature etc.).	
ASEAL A	--	No invasive/physical attacking tests required.
ASEAL B	--	No invasive/physical attacking tests required.
ASEAL C	IPA1	Simple attacks such JTAG attacks, memory dumps, simple I/O manipulations, offline attacks, or insider attacks (i.e. w/ user/owner privileges).
ASEAL D	IPA2	SYF1 + extended physical and invasive attacks such as side-channel attacks, fault injections, and complex manipulations (e.g. connection cutting, re-wiring, re-fuse), and readouts (e.g. micro-probing, optical memory read-out).

The above tables of course are only first high-level descriptions to demonstrate the general idea. A final ASEAL approach however would (in contrast to Common Criteria) (i) provide very concrete details and security requirements from typical automotive security risk cases and (ii) restrict minimum security requirements to protect against all reasonable automotive security attacks, but without trying to protect against all feasible automotive security attacks for instance by intelligence services or cyber war actors with virtually unlimited resources.

7 Conclusion & Open Challenges

This paper shows the strong need for systematic automotive security evaluation and discusses different methods for theoretical and practical security evaluation of automotive IT components. We focused especially on practical security testing for automotive components such as automotive penetration testing, since practical security testing is still relatively new to the automotive domain.

We believe theoretical and practical security evaluations will become standardized and mandatory procedures, similar to safety testing in ISO 26262 [27], to fulfill state-of-the-art product liability requirements and to protect various upcoming automotive business models (e.g. on pay per use basis). This of course requires serious efforts from OEMs, suppliers and security experts to establish necessary automotive security testing expertise, testing standards and testing infrastructures.

Moreover, we propose a standardized security evaluation process which provides four different assurance levels (ASEAL). Using ASEAL it is possible to assure that a specific automotive IT system has been security tested to a certain defined level. This can help to ensure worldwide comparable minimum security protection levels depending on the respective security risks similar to today's automotive safety integrity levels (ASIL).

References

[1] C. Miller und C. Valasek, „Adventures in Automotive Networks and Control Units," DEFCON 21 Hacking Conference, 2013.

[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner and T. Kohno, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," in USENIX Security, San Francisco, CA, USA, 2011.

[3] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson und H. a. o. Shacham, „Experimental security analysis of a modern automobile," Security and Privacy (SP), 2010 IEEE Symposium on, 2010.

[4] E. Markey, "As Wireless Technology Becomes Standard, Markey Queries Car Companies about Security, Privacy," Press Release of the US Senator for Massachusetts, Massachusetts, USA, 23.12.2013.

[5] C. Miller und C. Valasek, „A Survey of Remote Automotive Attack Surface," 2014.

[6] A. V. Thiemel, M. Janke und B. Steurich, „Speedometer Manipulation – Putting a Stop to Fraud," ATZ elektronik worldwide Edition, 2013-02.

[7] EXTREMETECH, „Hack the diagnostics connector, steal yourself a BMW in 3 minutes," [Online]. Available: http://www.extremetech.com/extreme/132526-hack-the-diagnostics-

connector-steal-yourself-a-bmw-in-3-minutes]. .

[8] EXTREMTECH, „Hackers can unlock cars via SMS,“ [Online]. Available: http://www.extremetech.com/extreme/91306-hackers-can-unlock-cars-and-meddle-with-traffic-control-systems-via-sms.

[9] The Cavalry, 2014. [Online]. Available: https://www.iamthecavalry.org/.

[10] Battelle , „Annual Battelle Cyberauto Challenge,“ [Online]. Available: http://www.battelle.org/site/cyber-auto-challenge.

[11] ISASecure, „Embeded Device Security Assurance (EDSA),“ [Online]. Available: http://www.isasecure.org/ISASecure-Program/EDSA-Certification.aspx.

[12] EMVCO, „EMVCO,“ [Online]. Available: http://www.emvco.com/.

[13] International Organization for Standardization (ISO), „ISO/IEC 15408-3: Information technology -- Security techniques -- Evaluation criteria for IT security,“ 2008.

[14] certicom, „FIPS Validation: what is it?,“ https://www.certicom.com/index.php/fips-validation-what-is-it.

[15] National Institue of Standards and Technology (NIST), „"FIPS PUB 140-2: Security Requirements for Cryptographic Modules,“ 2007.

[16] M. Wolf and M. Scheibel, "A Systematic Approach to a Quantified Security Risk Analysis for Vehicular IT Systems," in Automotive - Safety & Security, Karlsruhe, 2012.

[17] CERT, „Secure Coding Standards,“ [Online]. Available: http://www.cert.org/secure-coding/research/secure-coding-standards.cfm?.

[18] SAFECode, „Fundamental Practices for Secure Software Development,“ 2011.

[19] D.Angermeier, A.Kiening, H.Seudié, T.Stodardt und M.Wolf, „Trust Assurance Levels of Cybercars in V2X Communication,“ in Workshop on Security, Privacy, and Dependability for CyberVehicles (CyCAR 2013) co-located with ACM CCS, Berlin, 2013.

[20] D. K. Oka, C. Vuillaume and T. Furue, "Vehicle ECU Hacking," in ASIACCS, Kyoto, Japan, 2014.

[21] The OpenSSL Project, [Online]. Available: https://www.openssl.org/.

[22] M. I. S. R. Association und M. I. S. R. A. Staff, MISRA C:2012: Guidelines for the Use of the C Language in Critical Systems, Motor Industry Research Association, 2013.

[23] „OpenVAS - Open Vulnerability Assessment System,“ [Online]. Available: http://www.openvas.org.

[24] DEJA VU SECURITY, „Peach Fuzzer platform,“ 2014. [Online]. Available: http://peachfuzzer.com/products.

[25] „The Heartbleed Bug,“ 2014. [Online]. Available: http://heartbleed.com/.

[26] heise, BMW ConnectedDrive gehackt, heise.de, 2015.

[27] International Organization for Standardization (ISO), „ISO 26262: Road vehicles - Functional safety,“ 2011.

[28] M. Wolf, Security Engineering for Vehicular IT Systems - Improving Trustworthiness and Dependability of Automotive IT Applications, Vieweg+Teubner Verlag, 2009.

[29] Alliance of Automobile Manufacturers, „Auto Cyber-Security: Continual Testing, Checks and Balances,“ July 2014. [Online]. Available: http://www.autoalliance.org/auto-innovation/cyber-security.