Security-Aware Development of Cyber-Physical Systems Illustrated with Automotive Case Study

Viacheslav Izosimov [1 2], Alexandros Asvestopoulos [3], Oscar Blomkvist [3], Martin Törngren [1]
[1] KTH Royal Institute of Technology, Stockholm, Sweden
[2] Semcon Sweden AB, Göteborg, Sweden
[3] Scania CV, Södertälje, Sweden

***Abstract* — We present a method for systematic consideration of security attributes in development of cyber-physical systems. We evaluate our method in development of commercial vehicles that were so far unreasonably excluded from automotive security studies (despite the great importance of commercial vehicles for the society). We have conducted analysis of a known zero-cost non-physical attack, fine-tuned to our commercial vehicle (a truck), and considered countermeasures within the development flow.**

***Keywords* — Automotive systems, embedded security, cyber-physical systems, complex attacks, development flow**

I. Introduction

One of the biggest showstoppers in cyber-physical systems is security. When everything becomes interconnected and when system borders are difficult to identify, managing security threats is a great challenge. In this paper, we take as an example, cyber-physical system in a commercial vehicle and study security implications. Recent examples show that vehicles can be remotely controlled, with their components exposed to reverse engineering [1]. A number of scary movies are published on YouTube, with a driver unable to control the vehicle. However, we have not seen any "real" remote attacks of this type yet (with the exception regarding "relay" attacks for unlocking and stealing the vehicle). The reason is that attackers are not clear about their incentives and it does require some non-straightforward effort to wirelessly connect and execute a dedicated attack on such a specific cyber-physical system as a car. While they are stealing money from the bank, hacking in into the banking system, or hacking corporate network to get insider information, the incentives are clear and the attack portfolio is well established.

With respect to a passenger car, the most valuable asset is the car itself. The situation changes drastically though for commercial vehicles, in particular, for trucks. While the vehicle itself is still a great asset, other assets start to show up. Trucks carry valuable goods, which value is often greater than the value of the truck itself and some specific goods can be of particular interest for groups of dedicated attackers.

Unfortunately, commercial vehicles, with trucks not being an exception, have not been part of security research so far. In this paper, we first would like to claim that commercial vehicles could be subjected for similar types of attacks. We have done investigation of applicability of recent attack techniques on commercial vehicles and found that they are applicable. Compared to passenger cars, we see clear incentives for an attacker to perform an attack on the commercial vehicle and, hence, both manufacturers of commercial vehicles and security researchers should consider the threat seriously. Due to reduced availability of trucks, compared to that of passenger cars, some attack scenarios, as the one in this paper, can be first fine-tuned on a passenger car before the actual launch on commercial vehicles. In this paper, besides demonstrating our experiments on the actual truck, we discuss ways to efficiently include security into a design flow of a vehicle manufacturer. We have chosen to utilize a requirement elicitation process for security requirements and have performed analysis using, in particular, attack trees.

The remainder of this paper is organized as follows. Section II presents our motivation. Section III presents related work. Section IV discusses evaluation of the attack scenario selected. Section V presents our update to the design flow to capture security aspects and Section VI illustrates this design flow on an example. Section VII presents our conclusions and directions for future work.

II. Motivation

In this work, we were interested to study practical feasibility of a potentially infrastructure-critical security attack. Trucks are a critical part of the infrastructure, providing up to 70% deliverables of goods. Many areas, such as rural areas, are totally dependent on services enabled by the truck transports. Replacing one truck with another is often difficult and, if possible, may require substantial effort to reload the goods. Recent report suggests [2] that, without trucks, in 7 days, most of normal functions of the society will be largely degraded (including hospitals, food distributions, and alike). Yet, trucks are not well studied with respect to security.

Our hypothesis was that research findings from the passenger car domain would be applicable even in the case of trucks. Thus, we have selected the most promising attack scenario that would not require direct physical intervention with the truck, with the attacker utilizing external attack surfaces. We have also considered that physical availability for the truck is less, compared to that of passenger cars. Thus, complete reverse engineering of a truck, compared to the passenger vehicle case

[3], can be less likely and attackers will, hence, try to re-use findings from reverse engineering of a passenger car, also assuming that this should be possible. From attack surfaces, we selected attacks with the driver's mobile phone, connected via Bluetooth. This attack should be relatively easy and anonymous. Mobile phone, "paired" to the vehicle, is a powerful device that the attacker can utilize, without the owner's/driver's knowledge.

Attacks on mobile phones are well studied and it should not be a big challenge for an attacker to install malicious application on the driver's mobile phone, using, for example, social engineering tricks, pretending that the application is doing something useful. Moreover, since the malicious functionality targets the truck platform and not the mobile phone itself, it is likely to be undetected by the app stores. First step would be to install this application on a mobile phone and study internal architecture of the truck. Second step would be to execute the attack, after infecting a number of vehicles.

Our goal in this paper was, first, to evaluate whether this attack is feasible on a truck and, second, to suggest a possible efficient strategy to deal with security threats in development.

III. Related Work

Many smart vehicles utilize Bluetooth to support mobile phone connectivity. A gateway is often used to provide connection between the internal electrical system of the vehicle and this external Bluetooth interface. Internal electrical system is often based on CAN (Controller Area Network) on-board communication protocol. With respect to mobile phones, there are many developers that are familiar with the development of mobile applications, nearly half of them. Bluetooth and CAN bus standards are rather well known, as well as are their weaknesses. For instance, a Denial-of-Service (DoS) attack is straightforward on a CAN bus [4], while Bluetooth can be exploited in several ways [5], e.g. by the buffer overflow, especially in the older versions of the protocol. Bluetooth-CAN gateway is part of an on-board embedded computer that can be compromised with means that are used to compromise majority of other embedded systems. If the final target, CAN bus, is reached, the consequences can be serious. For example, Koscher et al. [6] demonstrated the amount of damage that can be achieved after an adversary gains access to the internal CAN network. Valasek and Miller [1] recently hit the news and social media, with the latest update on the remote control through telematics [7]. The key enablers of this "functionality" are the vehicle connectivity and advanced driver assistance systems (ADAS). In general, a great overview of possible attack scenarios can be found in [4]. In [3] attacks are enabled via extensive reverse engineering of passenger car components, with the following remote access to the vehicle via a whole variety of communication interfaces. There, authors already indicate that Bluetooth attack is one of the most promising.

Unfortunately, there is a lack of research publications on securing of commercial vehicles. With this work, we hope to close this gap, attempting to provide a possible solution to integration of security into development.

IV. "Non-Physical" Attack Scenario

Several steps comprise our attack scenario for reaching the internal CAN network of a commercial vehicle (see Figure 1). Driver's mobile phone is considered as an entry point, with the attack itself performed by a mobile phone application running on the phone. We have studied this attack scenario on both the actual commercial vehicle (to study consequences to the vehicle) and on the test bench (to have better controllability and observability for some of the attack steps). Note also that we have "bypassed" the

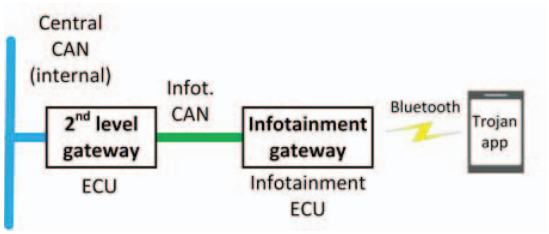

Figure 1. Attack scenario setup

infotainment system, leaving detailed investigation of the infotainment ECU (Electronic Control Unit) for future work, which is out of scope in this paper. The test bench included a breakout box, representing the internal and infotainment CAN segments in order to easily mount and connect various ECUs on the setup, along with the power supply units. The two main ECUs connected on the test bench were the two gateways (see Figure 1). The setup also included typical software and hardware tools by Vector [8] for analysis and stimulation of CAN bus communication. This allowed for injection of the malicious traffic and observation of its effect on the other CAN segments. Additionally, a laptop with a Bluetooth interface running was employed for testing the specific interface of the infotainment ECU using the various "open source" Bluetooth test tools available [9]. After testing in the test bench, we performed the same scenario on the actual truck to observe the flooding on the regular network load and to evaluate reaction of the truck driver.

In general, the attack scenario was feasible, which confirms that attack scenarios from passenger car domain can be transferred to commercial vehicles. Apart from the two gateway ECUs involved in the attack, 14 other ECUs were active and produced regular traffic with various shares of the bus bandwidth (see Table I). 10 out of 14 lost 5% or more of their share during the attack, 9 out of 14 lost 15% or more, and 8 out of 14 lost at least 50% of their share. The ECUs whose share of the bus dropped to zero, i.e. ECUs 1/3/4/5/8, appeared as shut down by the instrument cluster ECU, which also resides on this bus. Therefore, the instrument cluster filled up with warnings, quite a few of which were serious. Despite the overload, the truck was still possible to drive and safely park aside the road. Considering the warnings, the driver would pull up on the side of the road to call for assistance.

V. Security-Aware Design Flow

It is often argued by people with background in IT security that software of cyber-physical systems can be simply patched, with known mechanisms such as intrusion detection installed,

Table I. Traffic on the internal bus before and during flooding

Node	Regular bus share %	Attack bus share %	Change	Change % vs. regular
Inf. GW	0,11	82,22	82,11	74645
2nd GW	2	2,01	0,01	0
ECU 1	7,35	0	-7,35	-100
ECU 2	5,71	2,23	-3,48	-61
ECU 3	3,56	0	-3,56	-100
ECU 4	1,28	0	-1,28	-100
ECU 5	8,02	0	-8,02	-100
ECU 6	7,25	6,61	-0,64	-9
ECU 7	0,64	0,69	0,05	8
ECU 8	0,12	0	-0,12	-100
ECU 9	0,58	0,41	-0,17	-29
ECU 10	2,3	0,23	-2,07	-90

often ad hoc. For example, in [3], authors suggest disabling of vehicle functionality upon detection of malicious traffic. This is, unfortunately, not possible in many of cyber-physical systems, with trucks not being an exception. If the functionality is disabled or if an error is introduced due to security modification, the system can be taken out of order or commit dangerous behaviour. Every modification of safety-critical functionality has to be subjected to a so-called impact analysis, to evaluate that it does not potentially cause any dangerous behaviour.

In this work, we carefully modify the present design flow that was successfully used for integrating safety into the system. The first question is when the security should come in. The manufacturer should take care of security right from the beginning considering security requirements as part of the main requirement document, and not separately. This will enable balancing of security requirements against other requirements and enable review of the chosen security solutions for a security expert. The security chapter should be connected via traceability links to the respective subsystems' security requirements, following principles that we suggest in [10]. If it is too late, namely when a "zero-day" exploit is already reported, ad hoc emergency patching may have to be temporarily done to stop (or reduce implications) of on-going security attack. The work should then continue to properly integrate patching functionality, considering even modification of the requirements. Safety-critical parts of the vehicle should not be "patched" before scrutinizing required by the safety process. Otherwise, it may cause more problems than actually solving.

Except the "zero-day", security work should start by analysis of threats. One interesting approach is suggested in [11], where authors propose SAHARA methodology to include security analysis into safety hazard analysis and risk assessment (HARA) of automotive vehicles. SAHARA framework builds upon the HARA and ASIL analyses of ISO 26262, and combines them with the security threat model STRIDE. Each of the identified security threats for the system is categorized in terms of resources and knowledge required, and the threat's criticality. These three metrics have their own separate scales, and are combined to determine the security level index (SecL). Security threats with a SecL of the maximum value are potential safety hazards and should be included into the HARA process.

After analysis step, security requirements should be produced with the respective security-related failures (threats) indicated. We suggest the attack tree analysis. With this approach, it is possible to link the attack success to attack steps required to reach the final attack goal as well as to balance countermeasures with respect to cost/effect. The impact (or change) analysis should be also conducted for evaluation of mechanisms that may concern system performance and safety. For example, disabling of traffic on the internal communication bus will not be accepted neither will be accepted self-blocking of components. Here, we suggest usage of the event-tree analysis, where the event, triggered by security mechanism, propagates through system components.

In [3], it is pointed out that the integration between the system and its components is critical from a security point of view. In fact, the integration testing strategies have been traditionally considered as critical for the system even without security in mind. Today, these testing strategies have to be complemented with self-hacking and penetration testing, as an everyday practice.

VI. Illustration of the Proposed Method

Let us illustrate the proposed method on the example of the Bluetooth-mobile phone attack. The first step is to perform the SAHARA analysis, to determine SecL [11]:

Bluetooth-mobile phone attack: high security and possible safety relevance (T3); no prior knowledge on the truck required (black-box) (K0); and no additional tool required (R0). Thus, we obtain SecL of 4 (highest criticality).

Hence, the Bluetooth-mobile phone attack is rather critical and has to be addressed. Next step is to derive top-level requirements or "security goals", which, in turn, will be transformed into "security requirements". See Table II. Although security requirements can be constructed directly from SAHARA, requirement elicitation can benefit greatly from representing the attack as an attack tree, see Figure 3. An attack tree gives a great understanding on the alternative attack steps, with weights of the steps as a factor of risk. Attack trees [12] are tree-like structures where the root is the attack target, leaf nodes are attacks, and all intermediate nodes are either "OR" alternatives towards the same goal, or "AND" steps towards the node's parent goal. Note that, in our approach to attack trees, we use the CAND gate, or a conditional AND, proposed by Vavoulas et al. [13]. In the attack tree, the leaf nodes are assigned with the values, usually representing probability, to express the risk of an attack.

We can, with attack trees, technically analyse the attack steps one-by-one considering obtained effects and evaluating possible consequences for the system performance. First of all, we have two alternatives paths (corresponding to two CAND gates, respectively). We should focus on the most "risky" parts. In the case of the first path, N2: "Social engineering" is the one that we can work with first. Unfortunately, we cannot do much in the technical part but we should inform the driver of possible consequences of installing suspicious apps on the "paired" mobile phone in the truck manual. N1: "Trojan app" is again outside of our direct control but we should inform the driver about Trojan apps in the manual. N3: "Bluetooth service scan" is the technical one. We create requirement on restriction of Bluetooth profiles. As it is a technical mean, we should check with possible consequences for the system. Restrictions of the profiles would not lead to safety-critical problems and, hence, can be accepted. Let us now take the second path. In case of N5: "Social engineering to get documentation/source code", we cannot do much, except informing the IT department that this part of documentation is critical and should be additionally protected. With respect to N4 and N6, however, both related to Infotainment OS, we can apply the requirement FSR3. The event tree analysis against this measure confirms that integrity protection for the Infotainment does not lead to any safety violations and, hence, is acceptable. By considering integrity protection of the infotainment system, we can also address concerns for the attack step in nodes N9 and N10. Thus, integrity protection of the infotainment against manipulations is an efficient measure against the Bluetooth attack.

Table II. Requirements elicitation against Bluetooth attack

Req. type	Requirement text
SG	No inappropriate remote access to the internal vehicle CAN bus shall be possible through the Bluetooth.
FSR1	CAN bus shall be protected with gateways blocking unspecified ingress and egress traffic from/to external Bluetooth interface.
FSR2	CAN bus shall have active monitoring system for detection of suspicious internal traffic.
FSR3	Integrity protection shall be provided for parts of Infotainment concerned Bluetooth interface and internal CAN communications.

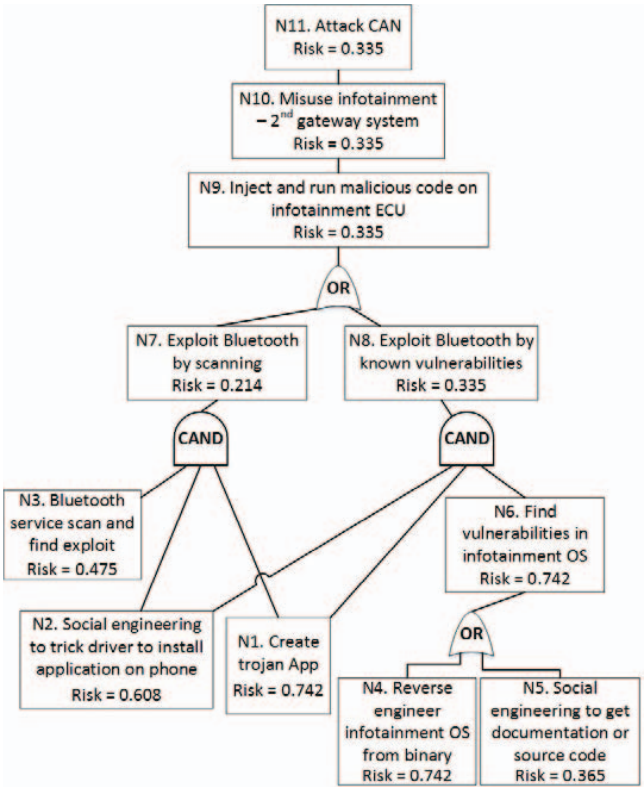

Figure 2. Attack tree for the attack scenario

Two other FSRs, FSR1 and FSR2, although derived initially, contribute only to the last level of defence needed for node N11 with the risk of only 0.335. In the design of security mechanisms, it is advisable to have more than one defence level. Thus, taking into account a low risk level of N11, we can choose either FSR1 or FSR2 to continue. Applying the event-tree analysis, we can see that blocking of ingress/egress message can potentially result in blocking of important commands to the network cluster. The active monitoring system, on the other hand, only reports issues detected to the driver. Thus, in the case with the first level of defence broken, it can be reasonable to keep the warning ready. However, if the infotainment unit cannot be protected from reverse engineering, then both intrusion detection and traffic blocking should be considered for the CAN bus, yet this defence measure will be rather weak in this case.

VII. Summary

In this work, we have addressed security aspects of cyber-physical systems on the example of commercial vehicles, trucks. Trucks can become a clear attack target from many angles and many of the "findings" from the passenger car domain can be reused. We have advocated for a careful analysis of security countermeasures at technical level before implementing them. It can both increase efficiency and reduce the unnecessary work.

As part of our future work, we would like to continue analysis on other attack scenarios for commercial vehicles, including attacks on autonomous vehicles and platooning.

References

[1] C. Valasek and C. Miller, "Adventures in Automotive Networks and Control," White paper, DARPA, 2013.

[2] "A Week without Truck Transport: Based on a Study Done by the Swedish Association of Road Haulage Companies in 2009," [Online]. Available: https://www.iru.org/cms-filesystem-action?file=mix-publications/week-without-trucks.pdf. [Accessed 18 09 2015].

[3] 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 Conf. on Security (SEC'11)*, 2011.

[4] P. Kleberger, T. Olovsson and E. Jonsson, "Security Aspects of the In-Vehicle Network in the Connected Car," in *IEEE Intelligent Vehicles Symposium*, 2011.

[5] P. Braeckel, "Feeling Bluetooth: From a Security Perspective," in *Advances in Computers*, vol. 81, 2011, pp. 161-236.

[6] Koscher, Karl; Czeskis, Alexei ; Roesner, Franziska ; Patel, Shwetak ; Kohno, Tadayoshi ; Checkoway, Stephen ; McCoy, Damon ; Kantor, Brian ; Anderson, Danny ; Shacham, Hovav ; Savage, Stefan, "Experimental Security Analysis of a Modern Automobile," in *IEEE Symp. on Security and Privacy (SP)*, 2010.

[7] "Chrysler Recalls 14m Vehicles after Jeep Hack," [Online]. Available: http://www.computerworld.com/article/2952186/mobile-security/chrysler-recalls-14m-vehicles-after-jeep-hack.html. [Accessed 18 09 2015].

[8] Vector, "CANalyzer," 2015. [Online]. Available: http://vector.com/vi_canalyzer_en.html.

[9] "Pwnie Express - Github," 2012. [Online]. Available: https://github.com/pwnieexpress/pwn_plug_sources/tree/master/src/bluetooth. [Accessed 08 May 2015].

[10] V. Izosimov, U. Ingelsson and A. Wallin, "Requirement Decomposition and Testability in Development of Safety-Critical Automotive Components," in *SAFECOMP*, 2012.

[11] G. Macher, H. Sporer, R. Berlach, E. Armengaud and C. Kreiner, "SAHARA: a Security-Aware Hazard and Risk Analysis Method," in *DATE*, 2015.

[12] B. Schneier, "Attack Trees," *Dr. Dobb's Journal,* vol. 24, no. 12, pp. 21-29, 1999.

[13] N. Vavoulas and C. Xenakis, "A Quantitative Risk Analysis Approach for Deliberate Threats," *Critical Information Infrastructures Security,* pp. 13-25, 2011.