

Formal Models for Automotive Systems and Vehicular Networks: Benefits and Challenges

Eduardo dos Santos
Department of Computer Science
University of Oxford
Oxford, OX1 3QD, United Kingdom
Eduardo.dosSantos@cs.ox.ac.uk

Dominik Schoop
Esslingen University of Applied Sciences
73732 Esslingen am Neckar, Germany
Dominik.Schoop@hs-esslingen.de

Andrew Simpson
Department of Computer Science
University of Oxford
Oxford, OX1 3QD, United Kingdom
Andrew.Simpson@cs.ox.ac.uk

Abstract—Formal models have seen widespread use in the development of safety- and security-critical systems — primarily as a means of providing increased assurance. In particular, formal models of threats have the potential to give rise to numerous benefits: they can help in the understanding of vulnerabilities (and the communication of such); they also provide a means by which these vulnerabilities can be reasoned about. We give consideration to how formal models of threats might be beneficial in the development of modern automotive systems and vehicular networks, which are increasingly interconnected. To achieve this we present a formal threat model of an idealised system. We use Predicate/Transition (PrT) Nets, a graphical mathematical formalism, to represent threats and plausible attacks. In addition to understanding and analysis, the approach also has the potential to provide benefits to the design and testing of vehicles’ distributed IT systems — thus contributing to an improved sense of security, privacy and safety. To this end, we give consideration to the challenges that lie ahead in terms of adopting such an approach.

Keywords—Formal Threat Model; Automotive Bus Systems; Vehicular Architectures; Automotive Security

I. INTRODUCTION

Equipping cars with networking features can give rise to a number of security and privacy threats [1] [2]. Moreover, these problems can assume dramatic proportions when one considers the fact that cars themselves will be connected with other cars in the near future [3] — at that point there will be little that distinguishes between external and internal communication. The motivating scenarios for such developments are numerous, being drawn from, for example, law enforcement and emergency transportation. There is, however, the clear potential for external communication (such as from Vehicular Ad-hoc Networks (VANETs)) to influence internal communication of such a connected vehicle — and, as such, its behaviour [4].

It follows that this extended environment has the potential to give rise to novel threats to security and privacy — and, ultimately, safety. For example, in our VANET scenario, control systems within the car might be accessible to adversaries in the VANET and could be susceptible to worm-like attacks spreading across the vehicular network. This, of course, is very different from the illustrations of

‘car hacking’ that we have grown used to, in which the security of an individual car is compromised [5].

In this paper, we seek to address a specific research question: “*how can security threats faced by connected vehicles be formally modelled?*” We will do this by presenting a motivating example. In addition, we will identify the benefits and challenges of embedding such an approach in the development process.

The formal modelling of threats can yield numerous benefits. First, it can help to gain a more precise understanding of the threats faced by the system under consideration, as well as the capabilities and limitations of attackers. Second, it can be used to help underpin several engineering activities, with risk assessment, automatic test case generation, and system validation and verification being specific examples [6]. Third, the combination of these models and the aforementioned activities have the potential to verify compliance with certain requirements [7] — or, at least, provide increased assurance.

In our domain of interest — vehicular networks — such an approach has the potential to aid understanding of threats, both in terms of a vehicle as a discrete unit and as part of a wider environment, such as a network of vehicles. This is of particular importance given the heterogeneity of the vehicular environment, characterised as it is by, for example, the different types of vehicles (e.g. bicycles, trams, cars and trucks) — developed by a variety of manufacturers — all of which have to co-exist in harmony. Future vehicular connectivity will make integration between all the possible single instances of vehicles a significant challenge.

Our work is motivated by the desire to assist those involved in the development of interconnected vehicles counter the increased threat landscape. In particular, we believe that the formal modelling of threats has a role to play in this regard — with this contribution being but a first step on a long journey.

The remainder of this paper is organised as follows. Section II provides the essential background information and contextualises our contribution. Section III summarises some attacks on automotive systems and vehicular networks. Section IV introduces Predicate/Transition nets, the formalism we adopt for our illustrative threat model, which

E. dos Santos is funded by the Science without Borders programme of CAPES Foundation/Brazil (grant no. 1029/13-4).

is shown in Section V. Then, Section VI shows an illustrative example of the formal modelling of threats. Next, in Section VII, we discuss the advantages of formally modelling threats relating them with the threat model we have developed. Finally, Section VIII summarises the contribution and plots the way ahead.

II. BACKGROUND

The primary motivation for undertaking this research lies in the apparent lack of formal threat models conceived for the emerging domain of automotive systems and connected vehicles. Most of the studies carried out to date have been exploratory ones (e.g. [1], [8] and [5]).

While these studies have contributed to the practical understanding of the security risks and threats involved in modern vehicles and future vehicular networks, they typically fail to address the threat landscape from a more formal perspective — by which we mean a perspective from which the system and its environment (and consequent attacks) are modelled via mathematical formalisms. Some of the benefits of such an approach were outlined in Section I; a further benefit is that such representations can be created from widely deployed notations such as UML or SysML — meaning that such models can be associated throughout the development lifecycle. Although those notations suffer from a lack of formal semantics and limited expressivity, there it is possible to generate formal representations from them. (See, for example, the contribution of [9].)

We assume a broad scope: we consider threats arising both from automotive systems (internal to the vehicle) and from vehicular networks (external to it). We adopt this breadth of scope and distinguish between the two aspects as both internal and external aspects present distinctive security challenges. Of course, the intersection of the external and internal environments will give rise to its own threats, which can be different from the threats faced from each of the environments in isolation.

We now make a few points on terminology. When we write *vehicle*, we are implicitly referring to *road vehicles*, specifically *cars* or *automobiles* (the terms are synonymous). Likewise, *vehicular networks* refer to the network formed by road vehicles connected to each other while in transit. We do not consider networks of vehicles of other types of transport, e.g. sea or air.

There are three major interacting distributed systems for a modern car (as shown in Figure 1):

- 1) the automotive bus system(s) (e.g. LIN, CAN, FlexRay and MOST) connecting electronic control units (ECUs), sensors and actuators with each other;
- 2) the wireless communication systems enabling the vehicle to communicate with external components (including Remote Keyless Entry systems, electronic tolling systems and connected vehicles, where vehicles access backend systems — usually via cellular networks); and
- 3) vehicular ad-hoc networks (VANETs), where vehicles communicate with each other mainly to enable enhanced safety applications on the roads.

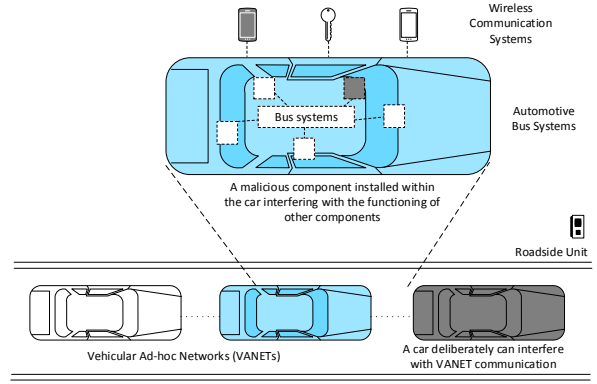


Fig. 1. High-level view of automotive systems and vehicular networks

These distributed systems, through ECUs and other components, interact to enable the full functionality of the car, e.g. external information from a backend system about a blocked road ahead will trigger the navigation system to plan an alternative route, or the fast deceleration of a car will be represented in the messages in the VANET.

Therefore, we see the potential for the formal modelling of threats in an integrated fashion. As in the re-routing example, there is a clear relationship between all of the distributed systems — and corresponding components — involved. If that relationship is misused, the security of the system as a whole can be jeopardised.

All of these distributed systems have to address certain high-level requirements to work in harmony. The need for those high-level requirements comes mainly from the characteristics of the environment into which the systems are inserted. An example of the influence of the environment is the importance of *time* in VANETs and automotive bus systems. For VANETs, time is a very important factor given the fast and continuous change of the network (due to the movement of cars). Similarly, automotive bus systems require real-time packet processing to minimise delays in communication between the different control systems in the car, thus contributing to enhanced safety guarantees.

We believe that, for a threat model to be considered of relevance and of practical application, it must take into account not only the unique characteristics of its target environment — by which we mean not only the environment the system is inserted into at a specific point in time, but also eventual changes to it over time.

Besides timing constraints, automotive bus systems also have a strong need for *scalability* and *adequacy to the lifecycle*. Scalability comes through the need for ensuring new added technology will behave correctly without adding new vulnerabilities or inadvertently changing the behaviour of the existing technology. Adequacy to lifecycle, in turn, indicates that existing technology needs to both be consistent within the lifetime of a particular instance of vehicle, and be consistent with older or newer instances of vehicles.

The application of formalisms into the automotive systems and vehicular networks has, so far, been limited. For example, Mundhenk *et al.* [10] employ a probabilistic

model-checking to analyse system-level automotive architectures with regards to security. As another example, Fazouane *et al.* [11] verify the consistency of privacy properties in a payment protocol for electric vehicles. To the best of our knowledge, we are the first to address the need for formally modelling security threats to vehicles and vehicular networks.

III. SUMMARY OF ATTACKS ON AUTOMOTIVE SYSTEMS AND VEHICULAR NETWORKS

In this section we review some attacks (potential and reported) against automotive systems and vehicular networks. Our intention is not to provide an exhaustive review of the literature; rather, we aim to provide sufficient references to motivate and frame the contribution.

Obviously, an attacker having access to the communication medium of an automotive system can carry out the four general attacks of interception (reading messages), interruption (stopping messages), modification (changing messages), and fabrication (generating new messages) — each of which will be useful in specific attack scenarios. In the following, we present one or two attacks for each of the three distributed automotive systems of Section II.

The value of the odometer determines when servicing a car is necessary; it also has some impact on the monetary value of the car. A car owner having access to the bus system (e.g. via the OBD-II interface) might want to reduce the value of the odometer to increase the car's value before sale [8].

Usually, a physical key is necessary to disable the immobilizer and to start the engine of the car. The immobilizer in the key lock executes a security protocol with the key fob when sufficiently close to the key lock (NFC) and the key turned in the lock then starts the engine [12]. In cars with keyless go systems, the engine can be started by pressing a button provided that the physical key is inside the car. It has been shown for some cars with keyless go that it is sufficient to simulate the presence of the key on the bus system with fabricated messages via OBD-II to start the engine [13].

Some car models enable their owners to control some functions via a mobile phone application, e.g. to turn on the heating but also to lock or unlock the car. In one such design, the car owner interacts with the mobile application that connects to a backend server to store the intended commands with the server. The backend server sends a cellular text message to the owner's car, triggering the car to connect to the backend server to collect the command — which is then executed by the car (e.g. unlock).

Obviously, an attacker who wants to have unauthorised inside access to a specific car can try to attack the owner's mobile phone. However, in some car models the communication protocol could also be attacked (e.g. models using the original BMW Connected Drive [14]). The attacker positions himself in the vicinity of the target car, fakes a radio base station, and sends a text message to the car indicating that a command is waiting at the backend server. The car will connect to a faked backend

server via the faked radio base station and will get a faked command from the attacker. The weakness is the command to authenticate against the car: providing the car's vehicle identification number (VIN) is sufficient. However, this VIN is not necessarily secret. Moreover, if there is an error in the communication with the attacker, the car will tell the attacker its VIN.

In a VANET, vehicles form an ad-hoc network and constantly broadcast information about their location, direction, speed, kind of vehicle, state of vehicle, and so on. A vehicle uses that information in addition to its sensor input to detect safety-critical road situations or to drive economically. In addition to the constantly broadcast cooperative awareness message (CAM), vehicles can send out a message containing information about a location (decentralised environmental notification message (DENM)). For example, a DENM can contain information about road conditions (icy patch, foggy, etc.) or about accidents. In contrast to the CAM, a DENM is routed using the geo-networking protocol (GN) to a target area where vehicles inside the relevant area might use the information in the DENM. An attacker could attempt to send fabricated CAMs making their car an emergency vehicle, thus getting green traffic lights and more space on the road. An attacker could attempt to fabricate a DENM faking an accident for a specific recipient vehicle causing the navigation system to reroute the vehicle to another road, which might be more suitable for the attacker to instigate a physical attack. An attacker could position himself to be in a GN-routing position and then drop selected messages to be routed (a gray hole attack), thus endangering other vehicles that do not receive the warnings. Such attacks, of course, should be prevented employing a PKI with anonymous certificates (pseudonyms).

IV. PREDICATE/TRANSITION NETS

Our threat model is developed using Predicate/Transition (PrT) nets — a type of high-level Petri net.

Petri nets are a graphical language used to model dynamic and distributed systems [15]. Although being a graphical language, Petri nets possess strong formal semantics [15]. Several extensions to Petri nets have been conceived over the years, including PrT nets [16], the main characteristic of which is the provision of dynamic features to the system being modelled. In other words, this means that the relationship between different system states can evolve as the behaviour of the system changes [16].

There are a number of advantages of using PrT nets for threat and attack modelling. In particular, their expressiveness allows the parallel representation of data and control flows [6], enabling the formal modelling of partially-ordered attacks [17] — a feature that other attack modelling techniques (e.g. attack trees [18]) fail to address satisfactorily.

Nevertheless, the most important reason for our choice of PrT nets is that they have already been used to foster security engineering activities. One example of the applicability of PrT nets is in the context of security testing. For example, Xu *et al.* use PrT nets to generate functional security test cases for web applications [6] and role-based

access control systems [19]. Furthermore, the graphical nature has the potential to facilitate its learning process as well as contribute to an easy visualisation of the system being modelled — thereby increasing accessibility. Moreover, the translation effort between other high-level modelling languages (such as UML) into PrT nets is, arguably, lower.

Our threat modelling method follows the approach described by Xu *et al.* [6]. It uses a slightly different definition of PrT nets from the one introduced by Genrich [16], although it keeps the formalism's basic notions.

Formally, a PrT net can be defined as follows.

Definition 1 (PrT net). *A PrT net is an 8-tuple, $N = \langle P, T, F, I, \Sigma, L, \gamma, M_0 \rangle$, where*

- P is a set of places (i.e., predicates);
- T is a set of transitions;
- F is a set of normal arcs with $F \subseteq P \times T \cup T \times P$;
- I is a set of inhibitor arcs with $I \subseteq P \times T$;
- Σ is a set of constants, relations (e.g., equal to and greater than), and arithmetic operations (e.g., addition and subtraction);
- L is a labelling function on arcs $F \cup I$, such that $L(f)$ is a label for arc f and each label is a tuple of variables and/or constants in Σ ;
- γ is a guard function on T , such that the guard condition of t , $\gamma(t)$, is built from variables and the constants, relations, and arithmetic operations in Σ ; and
- $M_0 = \bigcup_{p \in P} M_0(p)$ is an initial marking, where $M_0(p)$ is the set of tokens in place p and each token is a tuple of constants in Σ .

Consider Figure 2. We depict inhibitor arcs with dots at the ends. A double-pointed arc stands in for two arcs with the same label pointing in opposite directions. Two arcs with identical start and end are represented by one arc having two labels. Let the initial marking for the example be $M_0 = \{p_1(A), p_2(A), p_2(B)\}$, i.e. the token $\langle A \rangle$ is at places p_1 and p_2 and the token $\langle B \rangle$ is at place p_2 . A variable substitution $\theta = \{a/A, b/B\}$ assigns the constant A to the variable a and B to b . Given θ and the label $l = L(f)$ of an arc f , l/θ represents the substitution of all variables in l w.r.t. θ , i.e. given label $\langle a \rangle$, $\langle a \rangle/\theta = \langle A \rangle$. A transition t is enabled under θ if each place with an incoming arc to t holds a token matching the substituted label l/θ of the arc and there is no place with inhibition arc to t where the substituted label is a token of the place. In addition, the guard condition of t must be satisfied under θ .

In our small example, t is enabled assuming $A < B$ holds in Σ . When t is enabled under θ with the marking M_0 it can fire and generate a new marking M_1 , so we have the firing sequence $M_0 t \theta M_1$. The new marking removes those tokens from the incoming places that match the labels of the incoming arcs and adds those tokens to the outgoing places which are defined by the outgoing arcs. In our example, $M_1 = \{p_2(B), p_3(A, B)\}$.

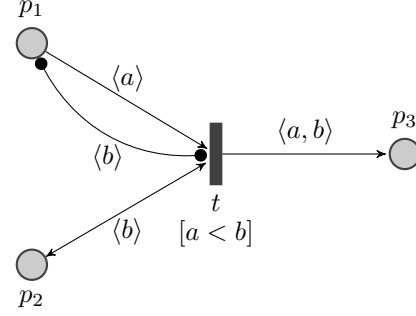


Fig. 2. A PrT net example

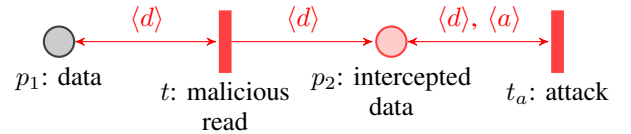
V. THREAT MODELLING

We now present our method for the formal modelling of threats in automotive bus systems and vehicular networks. We start by representing a representative, set of attacks at an abstract level using PrT nets (Section V-A); then we analyse some of the actors involved (Section V-B). Section VI gives examples of mappings to practical attack scenarios.

A. Attacks

Modelling attacks at an abstract level offers the potential for the re-use of modelled attacks across different parts of the system (or even across the different distributed systems involved in a modern car). Of particular importance are data-level attacks, which can happen at several moments of the data life cycle of a vehicle and can include internal and external data [20].

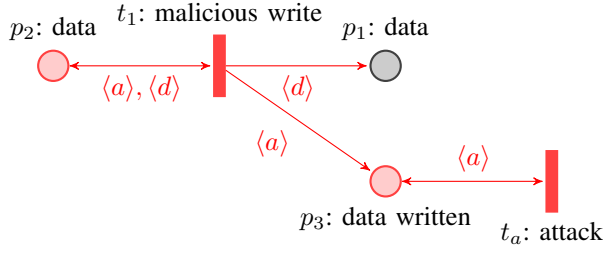
- 1) *Interception (unauthorised reading of data)*. Interception of data while in transit through any of the networks of the involved distributed systems causes loss of confidentiality (Figure 3). Here and in the following examples, the attack is represented by a firing sequence having t_a as the final transition.



Initial marking $M_0 = \{p_1(D), p_2(A)\}$

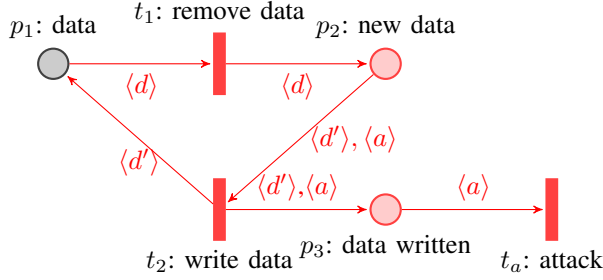
Fig. 3. Interception (unauthorised reading of data)

- 2) *Fabrication (unauthorised introduction of data)*. This attack refers to the unauthorized sending of data (spoofing), causing loss of authenticity. Injection and fabrication have the same meaning (Figure 4).
- 3) *Modification (unauthorised change of data)*. A data modification (or creation) attack leads to loss of integrity. Modification can also assume the form of creation of new data (Figure 5).
- 4) *Modification (unauthorised change of functionality)*. We idealise this attack as a special case of a data modification attack (Figure 6). In this instance, a whole component would be modified (or physically



Initial marking $M_0 = \{p_2(A, D)\}$

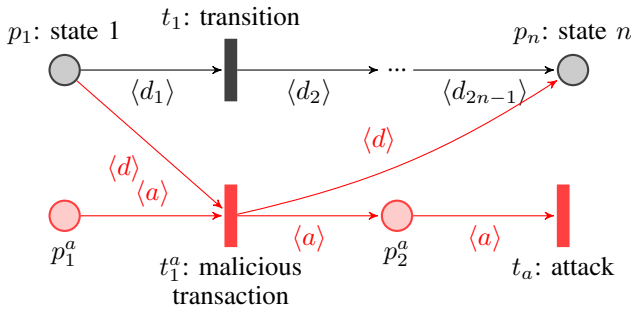
Fig. 4. Fabrication (unauthorised introduction of data)



Initial marking $M_0 = \{p_1(D), p_2(A)\}$

Fig. 5. Modification (unauthorised change of data)

replaced) by a modified copy of it (e.g to install remote surveillance capabilities). In the example net, the normal sequence of states is changed. Alternatively, loss of functionality can be caused by data modification.

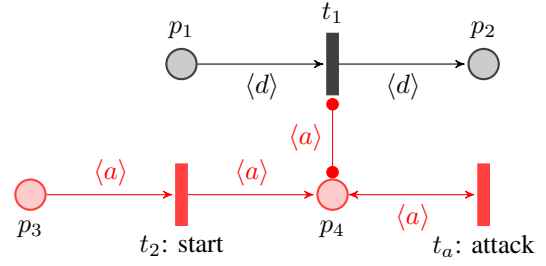


Initial marking $M_0 = \{p_1(D), p_1^a(A)\}$

Fig. 6. Modification (unauthorised change of functionality)

5) *Interruption (unauthorised stop of data transfer or program execution)*. This attack refers to the unexpected interruption of information flow within a network or the interruption of a specific component or program. It causes loss of availability. There are two ways to model an interruption attack in PrT nets: via inhibition or via data consumption.

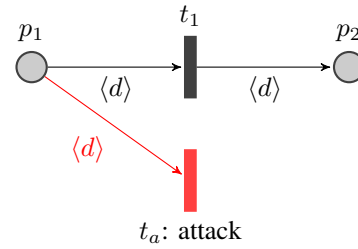
- Interruption via inhibition*. An interruption via inhibition is modelled with an inhibition arc (Figure 7). When transition t_2 fires, transition t_1 cannot fire any more (interruption).
- Interruption via data consumption*. An interruption via data consumption can be modelled with the



Initial marking $M_0 = \{p_1(D), p_3(A)\}$

Fig. 7. Interruption (unauthorised stop of data transfer or execution via inhibition)

necessary token required for t to fire removed by some malicious transition of the attacker (Figure 8).



Initial marking $M_0 = \{p_1(D)\}$

Fig. 8. Interruption (unauthorised stop of data transfer or execution via data consumption)

B. Actors

We now list the main actors involved in our threat model. Carrying out an in-depth analysis of all actors involved in the system falls outside the scope of our contribution; rather, our actors serve motivational and illustrative purposes. For a broader review of the topic, we refer the reader to [21] and [22].

- 1) *Driver*. Car drivers have unlimited access to the vehicle, which, in most cases, is their own property. They do not usually have knowledge about the internal functioning of a car. In some car models (e.g Tesla), drivers can install third-party software from the OBD-II port. Depending on the drivers' professional background, they might also have hacking/programming skills, which constitutes an incentive to development of personalised software for their own cars. Car drivers are not limited to the vehicle's legal owner or their relatives, but rather should include everybody allowed to drive the vehicle (e.g valets, car-sharers, etc.).
- 2) *Evil Mechanic*. An evil mechanic has unlimited access to different cars at the same time. They also have a deep knowledge of cars' internal functioning as well as specialist diagnostics equipment. As a result, they can easily modify cars' hardware and software components. In addition, they can also manipulate data stored within the vehicle [23]. Commonly, drivers have no way to tell what modifications their car underwent after being collected from the garage.

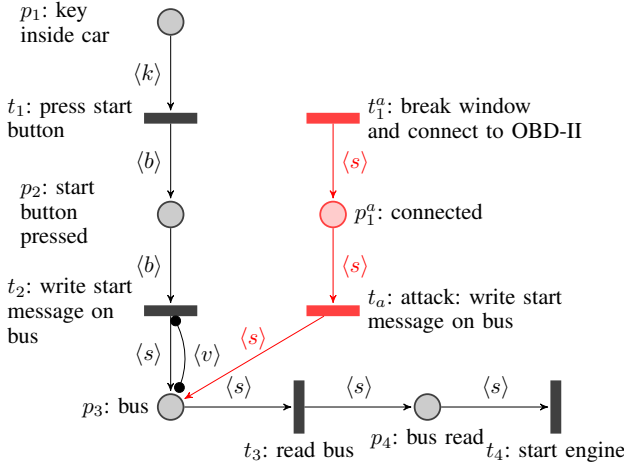


Fig. 9. Car engine start via OBD-II port attack

VI. ILLUSTRATIVE EXAMPLES

We now show how our method could be applied to model practical attacks, building upon the basic attacks of Section V-A. The following attacks are considered: engine start without key (Section VI-A), unauthorised modification of the odometer value (Section VI-B), and rerouting because of a faked warning (Section VI-C).

We have no intention to be complete here (due, primarily, to limitations on space); rather, we aim to provide sufficient evidence of the applicability of our method. Non-technical steps of attacks are included in the modelling for clarification purposes only: to describe a practical situation in which the attack could happen (e.g. “break window and connect to OBD-II”). We have no intention to expand our modelling to related domains, e.g. criminology — our attack models are designed to cover technical elements only. A further discussion about extensions and applicability to security engineering follows in the next section.

A. Engine start without key

In the first attack, a car thief starts the car engine via OBD-II socket instead of pressing the button of a keyless go system. This attack can be visualised in Figure 9. Given $\theta = \{s/START\}$ and any initial marking M , a firing sequence leading to an attack is $Mt_1^at_a\theta_3\theta_4$.

B. Odometer value modification

Next, the second attack addresses the unauthorised modification of the odometer value by accessing the OBD-II socket (chip tuning) [24]. Although we have restricted ourselves to the odometer value, the modification of any internal value could be re-used from this model with minor changes [8]. This attack is shown in Figure 10.

A possible firing sequence leading to an attack is $M_0t_{1,1}\theta_1t_{2,1}\theta_1t_{2,2}\theta_1t_{2,3}\theta_1t_{2,6}\theta_2t_{1,1}$ with

$$M_0 = \{p_{1,2}(OD), p_{2,1}(PWD), p_{2,3}(F_1, ND)\},$$

$$\theta_1 = \{d/OD, p/PWD, f/F_1, x/ND\}$$

and

$$\theta_2 = \{d/ND\}.$$

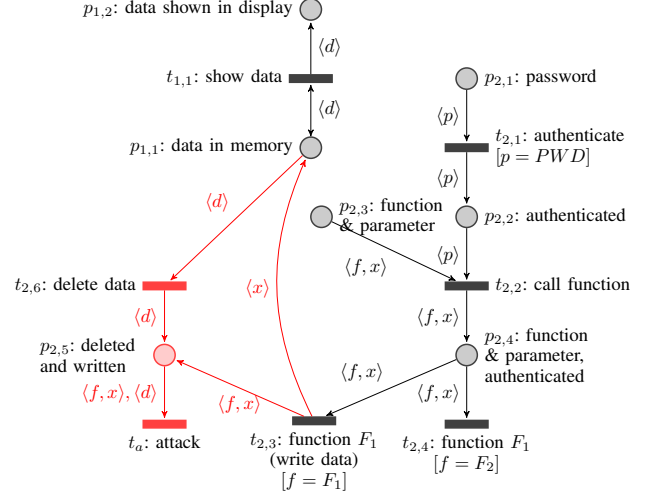


Fig. 10. Odometer modification attack

C. Rerouting due to faked warning

In this example, an attacker aims at getting physical access to a car and its passengers, possibly to hijack the car. Therefore, the attacker misuses the functionality of a VANET. She moves in the vicinity of the car to hijack and gets it to be driven to a more quiet location by sending a faked warning message [25] [26]. A PrT net modelling the attack is shown in Figure 11. Given $\theta = \{d/D, a/A\}$ and any initial marking M , a firing sequence leading to an attack is $Mt_1\theta t_2\theta t_3\theta t_1^a\theta t_2^a\theta t_4\theta t_5\theta t_3^a\theta t_4^a$.

VII. DISCUSSION

Formal threat models have been an active area of research in other fields, especially network protocols. The most notable example of them is probably the *Dolev-Yao* threat model [27], which establishes the *de facto* standard of protocol security in terms of attacker capabilities in a worst-case scenario.

Although widely-accepted as the *de facto* standard for network protocols, formal threat models like Dolev-Yao’s have limitations that, ultimately, undermine their applicability in the present context. Consequently, there is the need for the development of novel formal threat models specifically crafted for automotive and related systems.

We would argue that it is important for automotive systems to be developed in accordance with formal threat models. This is especially important given the emerging complexity of these systems. Even if components (such as vehicle ECUs) are designed to be secure, there still remains the potential for misuse given their need for integration with other components or systems within the vehicle [28].

Instead of attacking a component individually, attacking the interfaces between components is more likely to result in success [28]. That can have a severe impact on the security of automotive systems, given the speed with which new ECUs and sensors are added into automobiles,

The addition of new components can bring various benefits in terms of performance and interconnectivity, but

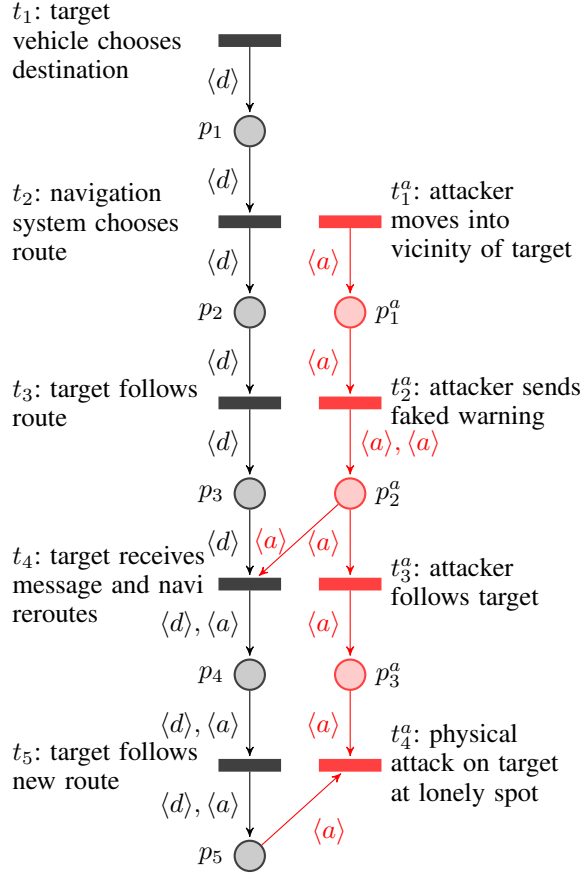


Fig. 11. Rerouting attack in VANET

can also introduce new vulnerabilities and risks. This has already been shown in practice: it was found that the addition of new cyber-physical features to a Jeep Cherokee 2014 introduced vulnerabilities non-existent in previous models of the car [5].

Given the complexity of automotive bus systems and their surrounding networks and systems (e.g vehicular ad-hoc networks and wireless communication systems), mitigation of these risks can hardly be achieved without the use of automated tools, which in turn require formal threat and architectural models of the systems involved.

We glimpse the potential of our PrT attack nets being used to finding vulnerabilities on existing systems. That would happen by creating PrT net models of the systems themselves. Given the large number of different systems and components in the car, system models need to be automatically obtained. One of the ways to do so is through the transformation of less formal UML diagrams into more formal representations. This is likely going to require UML diagrams that represent the dynamic behaviour of the system over time (e.g sequence diagrams or statecharts) given that it's the system's internal state we are concerned with. An example of transforming UML models to a formal language can be found in [29].

Having bridged the gap between system models and attack models, we can then start the automatic generation

of security test cases, in a similar way to the contribution of Xu *et al.* [6]. The attacks presented in this paper are just a starting point with scope for much more. If successful, attacks can be said to, likely, have found a vulnerability on the system. There might be the need to deal with false positives. Attack models are fired against system models, possibly in a random fashion. Both of these models are represented as test code in the generated test cases. There is the need to establish a correspondence between attack models and attack code, which will be the subject of future work. Currently, we see this approach working best during the software implementation stage of the development life cycle (when code is written and tested).

Furthermore, integrating security and safety is not an easy task. Nowadays, the boundaries between safety and security have become blurred [30]. The interaction between safety and security has already been noticed by the car manufacturers and suppliers; however, contrastingly with other domains, there is little work contemplating that interaction in a more formal fashion.

Our analysis of the literature has shown that there are few formal threat models for automotive and related systems. We believe that the best way to achieve an optimum sense of security aligned with safety is by developing solutions with a view to integration with vehicular networks from scratch. In short, security of both of these domains must be addressed in parallel to provide higher assurance levels.

Because of their levels of interoperability, one can visualise the threat environment of vehicular networks as being an extension to automotive systems. One should not underestimate the potential for integration in the near future. Existing applications in the intersection of the two domains may be only the starting point of a much wider development.

VIII. FINAL CONSIDERATIONS AND FUTURE WORK

We have presented a formal model of security threats faced by automotive systems and vehicular networks. The main motivation for our contribution comes from the lack of contributions addressing the security problems of the modern car under a more formal perspective. The formal models of threats and attacks idealised herein are the first step on a journey that, in the opinion of the authors, will bring more automation to the security engineering of the car — with security testing being the most noticeable example.

Although each of the involved networks (and, consequently, systems) cover different aspects, the strong relationship between them is clear. Vehicular networks can develop in a way that affects how a vehicle works internally. The number of components (e.g ECUs) existing in a car is already significant; connecting them to other external networks adds an unimaginable level of difficulty to guarantee all components work smoothly as expected. If a single component is misprogrammed, it may pose security threats not only to components immediately nearby, but threats may spread to other networks. This is facilitated by a business model in which components are developed by different manufacturers with little interaction between them. While

component integration standards do exist, they cannot tell us much about how secure a component's implementation is.

This work is part of a wider research project with the objective of developing a model-based framework to test the functional security of cars and associated systems. It has a particular emphasis on connected vehicles. When realised, we expect the framework to become a valuable tool in the exhaustive and automatic testing of automotive systems. Some of the questions we hope this framework will help to answer include: “*Can the infiltration of fake data on a component X compromise the behaviour of other components in the vehicle? To what extent?*”, and “*Do redundancy countermeasures work as expected and are helpful to guarantee the safety of the car?*”.

Although the paper deals with the problem at an abstract level by considering several possible attacks, in the longer term our framework will generate concrete testing code. It is worth noting that the list of attacks presented is not exhaustive: there is scope for many more to be added to the list. In fact, we also see potential in the modelling of network-level attacks, similar to what early exploratory studies on automotive security did (e.g. [1], [8]).

REFERENCES

- [1] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, and others, “Comprehensive Experimental Analyses of Automotive Attack Surfaces,” in *USENIX Security Symposium*. San Francisco, 2011.
- [2] L. Othmane, H. Weffers, M. Mohamad, and M. Wolf, “A survey of security and privacy in connected vehicles,” in *Wireless Sensor and Mobile Ad-Hoc Networks Vehicular and Space Applications*, 2015, pp. 217–247.
- [3] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, “VANET security surveys,” *Computer Communications*, vol. 44, pp. 1–13, May 2014.
- [4] C. Sommer and F. Dressler, *Vehicular Networking*. Cambridge : Cambridge University Press, 2015.
- [5] C. Miller and C. Valasek, “Remote Exploitation of an Unaltered Passenger Vehicle,” Tech. Rep., Oct. 2015.
- [6] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and W. Xu, “Automated Security Test Generation with Formal Threat Models,” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 4, pp. 526–540, Jul. 2012.
- [7] Y. Aoki and S. Matsuura, “Verifying security requirements using model checking technique for UML-based requirements specification,” in *Requirements Engineering and Testing (RET), 2014 IEEE 1st International Workshop on*, 26–26 Aug. 2014, pp. 18–25.
- [8] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and others, “Experimental security analysis of a modern automobile,” in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 447–462.
- [9] J. Jacobs and A. Simpson, “On the formal interpretation and behavioural consistency checking of SysML blocks,” *Software & Systems Modeling*, pp. 1–34, 2015.
- [10] P. Mundhenk, S. Steinhorst, M. Lukasiewicz, S. A. Fahmy, and S. Chakraborty, “Security Analysis of Automotive Architectures Using Probabilistic Model Checking,” in *Proceedings of the 52Nd Annual Design Automation Conference*, ser. DAC ’15. New York, NY, USA: ACM, 2015, pp. 38:1–38:6.
- [11] M. Fazouane, H. Kopp, R. van der Heijden, D. Le Métayer, and F. Kargl, “Formal Verification of Privacy Properties in Electric Vehicle Charging,” in *Engineering Secure Software and Systems*, ser. Lecture Notes in Computer Science, F. Piessens, J. Caballero, and N. Bielova, Eds. Springer International Publishing, 2015, vol. 8978, pp. 17–33.
- [12] R. Verdult, F. D. Garcia, and B. Ege, “Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer,” in *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14–16, 2013*, 2013, pp. 703–718.
- [13] Securemycar, “How cars are stolen through OBD port theft and key cloning,” Aug. 2014. [Online]. Available: <https://www.youtube.com/watch?v=dvmSOEKfkug>
- [14] D. Spaar, “Car, open thee! Vulnerabilities in BMWs ConnectedDrive,” *Magazin fur Computer Technik*, p. 86, May 2015, In German. [Online]. Available: <http://www.heise.de/ct/ausgabe/2015-5-Sicherheitsluecken-bei-BMWs-ConnectedDrive-2536384.html>
- [15] T. Murata, “Petri nets: Properties, analysis and applications,” *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
- [16] H. J. Genrich, “Predicate/Transition Nets,” in *Petri Nets: Central Models and Their Properties*, ser. Lecture Notes in Computer Science, W. Brauer, W. Reisig, and G. Rozenberg, Eds. Springer Berlin Heidelberg, Sep. 1986, no. 254, pp. 207–247.
- [17] D. Xu and K. E. Nygard, “Threat-Driven Modeling and Verification of Secure Software Using Aspect-Oriented Petri Nets,” *IEEE Transactions on Software Engineering*, vol. 32, no. 4, pp. 265–278, Apr. 2006.
- [18] B. Schneier, “Attack Trees,” <https://www.schneier.com/attacktrees.pdf>, 1998.
- [19] D. Xu, M. Kent, L. Thomas, T. Mouelhi, and Y. L. Traon, “Automated Model-Based Testing of Role-Based Access Control Using Predicate/Transition Nets,” *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2490–2505, Sep. 2015.
- [20] J. Petit, M. Feiri, and F. Kargl, “Revisiting attacker model for smart vehicles,” in *Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on*. IEEE, 2014, pp. 1–5.
- [21] R. R. Brooks, S. B. Yun, and J. Deng, “Chapter 26 - Cyber-Physical Security of Automotive Information Technology A2 - Das, Sajal K.” in *Handbook on Securing Cyber-Physical Critical Infrastructure*, K. Kant and N. Zhang, Eds. Boston: Morgan Kaufmann, 2012, pp. 655–676.
- [22] B. Mokhtar and M. Azab, “Survey on Security Issues in Vehicular Ad Hoc Networks,” *Alexandria Engineering Journal*, vol. 54, no. 4, pp. 1115–1126, 2015.
- [23] J. Petit and S. E. Shladover, “Potential Cyberattacks on Automated Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, Apr. 2015.
- [24] Instructables, “Odometer Reprogramming,” <http://www.instructables.com/id/Odometer-Reprogramming/>, 2016.
- [25] M. Raya and J.-P. Hubaux, “Securing vehicular ad hoc networks,” *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, Jan. 2007.
- [26] D. Symeonidis, “RDS-TMC Spoofing using GNU Radio,” in *Proceedings of the 6th Karlsruhe Workshop on Software Radios*. Karlsruhe University, 2010, p. 6.
- [27] D. Dolev and A. Yao, “On the security of public key protocols,” *Information Theory, IEEE Transactions on*, vol. 29, no. 2, pp. 198 – 208, Mar. 1983.
- [28] F. Sagstetter, M. Lukasiewicz, S. Steinhors, M. Wolf, A. Bouard, W. Harris, S. Jha, T. Peyrin, A. Poschmann, and S. Chakraborty, “Security challenges in automotive hardware/software architecture design,” 2013, pp. 458–463.
- [29] D. Varró, M. Asztalos, D. Bisztray, A. Boronat, D.-H. Dang, R. Geiß, J. Greenyer, P. V. Gorp, O. Knemeyer, A. Narayanan, E. Rencis, and E. Weinell, “Transformation of UML Models to CSP: A Case Study for Graph Transformation Tools,” in *Applications of Graph Transformations with Industrial Relevance*, ser. Lecture Notes in Computer Science, A. Schürr, M. Nagl, and A. Zündorf, Eds. Springer Berlin Heidelberg, Oct. 2007, no. 5088, pp. 540–565.
- [30] L. Piètre-Cambacédès and M. Bouissou, “Cross-fertilization between safety and security engineering,” *Reliability Engineering & System Safety*, vol. 110, pp. 110–126, Feb. 2013.