Enabling Efficient Common Criteria Security Evaluation for Connected Vehicles

Angelos Stamou*, Panagiotis Pantazopoulos*, Sammy Haddad‡ and Angelos Amditis*
* Institute of Communication and Computer Systems (ICCS)
Iroon Polytechniou Str. 9, GR-15773, Athens, Greece
Email: {angelos.stamou, ppantaz, a.amditis}@iccs.gr
‡ Oppida
6, Avenue du Vieil Etang, 78180 Montigny-le-Bretonneux, France
Email: sammy.haddad@oppida.fr

***Abstract*—Cyber-security assurance evaluation seeks to gain evidence that the relevant requirements of an IT system are met. Towards that end, carefully-designed evaluation processes of the considered systems are needed. The only so-far validated approach, the Common Criteria (CC) standard, relies on exhaustive evaluation tasks to provide (up to) the highest possible assurance at the expense of increased costs. When the evaluation involves the connected vehicles paradigm which integrates a mosaic of third-party modules and interfaces, applying CC becomes problematic; the cost in resources and time further increases while relevant automated tools or document templates, are scarce.**

This paper introduces the AFT (Assurance Framework Toolkit) which is a platform-independent online software toolkit that *enables efficient* CC-based cyber-security evaluations on products of the automotive cyber-physical ecosystem. A set of relevant CC-specific security assurance needs are explained and the way that the AFT software-design and functionality covers them, is presented. Subsequently, the development of the toolkit (with publicly available source-code) as well as its capability to meet the evaluation of automotive needs, are detailed. Finally, an empirical study estimates the expected AFT gains against typical CC unassisted evaluations. The proposed toolkit (along with its extendibility feature) practically tackles the cost-limitations of standardized security evaluations filling an important technology gap towards safer connected driving.

I. Introduction

One of the most critical parameters that would determine the adoption and effective market-penetration of the connected vehicles paradigm *i.e.*, the cyber-physical system of highly-equipped and infrastructure-connected vehicles, is expected to be its reliability [4]. It follows that the largest the trust in the involved technology the easiest the adoption of connected and (highly) automated driving. This rule is reflected in the ever-increasing cyber-security interest and largely explains its particular importance for connected vehicles [10]. Apart from the design and introduction of cyber-security (preventive) mechanisms [12], a subsequent objective is to establish procedures providing *assurance* that the connected vehicles ecosystem satisfies its intended cyber-security behavior.

The general problem of Information Technology (IT) security assurance [1] has been an open research thread along the last four decades or so. Till now, there have been different approaches that seek to address three main dimensions placing varying importance on them: the product to be evaluated (together with its environment and involved threats), the type of applied evaluation processes (and their focus) and finally, the required evaluator's profile and expertise. Each approach exhibits certain pros and cons.

The first approach *i.e.*, the conformity checks, evaluate a system's compliance to a specific reference expressed in a conformity list [13]. The main limitations amount to the definition, maintenance and keeping up-to-date of the list while anything not conformant to (a part of) the list cannot be validated. On the other hand, conformity checks provide typically the fastest and lowest-cost evaluation achieving medium levels of confidence (in the product's security); evaluation results are easy to understand and compare since they are the same for every product evaluated. The second approach *i.e.*, vulnerability tests defines an evaluation perimeter specifying the product, the tests environment and associated limitations [5]. An expert conducts any test of her choice that is in-scope (*i.e.*, perimeter) in a predefined time. The result is a set of potential vulnerabilities identified by the (specific) tester and therefore, the method provides low to medium assurance level. Results are obtained relatively fast but are not directly comparable since different testers may select different tests (for the same product). Another disadvantage is the strong reliance to the tester's competence.

Assurance frameworks constitute the most complete and exhaustive approach. They provide the highest possible assurance levels at the expense of increased cost and time. They also require the involvement of rare accredited evaluators. The relevant background is rather limited to the NIST FIPS 140-X [6], the US Trusted Computer System Evaluation Criteria TCSEC [7] and the European IT Security Evaluation Criteria ITSEC [8]. The flagship ISO Common Criteria (CC) for IT Security Evaluation 3.1 R5 [3], a merging of the two aforementioned ones, is the only evaluation standard officially recognised by more than 30 countries worldwide [15]. CC defines a framework for the oversight of cyber-security evaluations, an implementation-independent syntax (called Protection Profile) for specifying the security requirements to be met and a comprehensive methodology for evaluating them. Independent CC evaluators examine thoroughly both the security functional requirements and the evidence (*i.e.*, subject to the targeted assurance level) that those requirements are correctly implemented. Most of the examined documents need to have been earlier prepared by the product developer.

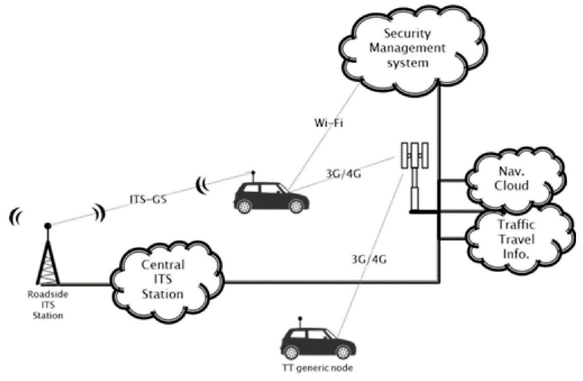


Fig. 1. The introduced toolkit's focus (*i.e.*, so-far included data and dedicated features) is on vehicle-to-infrastructure (V2I) communications instances whereby the vehicle connects to roadside stations and cloud services

Such a CC evaluation includes the gathering of an exhaustive set of data *i.e.*, inputs to eight different assurance classes that the CC framework requires. Each class is further decomposed to several families covering the identification of the evaluated system's called Target of Evaluation (ToE) assets, involved threats, assumptions, security objectives, functional design, guidance documents, suitable security testing and many others. Especially when the ToE is the connected vehicle, collecting such data is a demanding and time-consuming task requested from the ToE's developer. Further difficulties emerge as CC is not tailored for the automotive setting, the relevant technologies are continuously evolving while most developers have hardly any security evaluation experience.

To practically address those challenges and minimize the involved costs of a CC-based evaluation having the connected vehicle (with the infrastructure, see Fig. 1) as the ToE, we introduce the AFT *i.e.*, a *platform-independent* and *open-source*[1] toolkit; it facilitates the cost-efficient compilation of the required evaluation inputs for the security target evaluation (ASE class), the architecture and functional specification reviews (ADV class) as well as the expected functional tests (ATE class) [3]. The toolkit has so-far received automotive data-inputs from a careful identification of implementation-independent security requirements for connected vehicles (*i.e.*, the Protection Profile) we have previously introduced [2] as well as our relevant testing results over a real-world V2I testbed [11]. Importantly, AFT can be extended to cover other -less demanding- CC families, if needed. In conclusion, the toolkit leverages CC-based evaluations of automotive products by addressing their cost limitations (in time and resources) and thus, constitutes an important contribution for the security practitioner (automotive) community.

The remainder of the paper is structured as follows: Section II presents the cyber-security evaluation background and relevant software tools. Section III details the AFT specifications and software design; subsequently, the AFT implementation is explained. Section V empirically studies the expected cost-reduction that AFT can achieve and Section VI concludes the paper.

[1]AFT code can be downloaded from: https://isense-gitlab.iccs.gr/safertec/aft

II. Security Evaluation Background and Relevant Tools

Contrary to conformity checks and vulnerability tests that provide (up to) medium assurance levels, connected vehicles call for the highest possible assurance, as automotive cybersecurity may affect functional safety [12]. To achieve that, a comprehensive description of security requirements and the execution of an exhaustive list of evaluation tasks is employed by CC. It follows that the involved costs are increased, potentially hindering the approach's applicability; a relevant evaluation project (of a given product) typically takes about 12 months to complete requiring up to USD $ 100K [16].

Two are the main means to reduce costs; one is to adapt the CC framework and its needs to a certain domain (*e.g.*, the automotive ecosystem) while the second and presumably most effective one, is to introduce dedicated to that domain, software tools that propose default values and help gathering the required input-data automating the evaluation. Along the first front, no more than two approaches have been introduced in literature to adapt CC to the specific needs of connected vehicles. The CARSEM approach proposed the parallelization of evaluation tasks and their distribution among different actors [14]. The SAFERtec project [23] framework further enhanced CARSEM introducing a knowledge base for automotive security assurance as well as proposing evaluation methods at system-level [1].

Common Criteria and hence, the above approaches start from the detailed identification of the security environment (*i.e.*, assets, threats, assumptions *etc.*), understand in detail the system under analysis *i.e.*, Target of Evaluation (ToE) and subsequently describe in a systematic way its security objectives and security requirements. Towards that end, two important documents are defined (in [3]): The Protection Profile (PP) and the Security Target (ST). Both adopt a certain structure and terminology to formally define the involved security functional requirements (SFRs) and security assurance requirements (SARs). The difference is that PP describes requirements that are implementation-independent while ST refers to one specific ToE implementation.

The ToE assurance evaluation relies on a broad set of evaluation tasks (to be carried-out) categorized under 8 security assurance classes with each class having several families [3]; the achieved assurance level depends on both the width (*i.e.*, number of assurance families used) and depth (*i.e.*, usage-level of components for that family) of the analysis for each class. Three fundamental evaluation classes are currently supported by AFT: a) *ASE*: the main document is the ST that specifies the functions under evaluation and the assurance requirements to be met; it also determines the rest of the required evaluation inputs; b) *ADV*: the evaluation of design documents. It includes detailed design specs and source code reviews as well as documents describing the ToE interfaces and their relation to the SFRs; c) *ATE*: it includes the functional tests. The ToE developer describes a test plan including test scenarios or test scripts. For each scenario, the prerequisites, operations and expected results need to be detailed (to be subsequently used by the evaluators in order to confirm the operation of the security functions). Those classes are typically supported by the set of software tools assisting CC-based evaluations.

This set is indeed quite limited; CCMODE is a solution [17] which allows to produce all the necessary CC documents and supplementary documents but requires certain management for the development-environment configuration (named EMT). In [18] the GEST automatic generator of security target templates from already evaluated and certified products was presented. TL SET software was introduced by Trusted Labs [20] to act as a smart Security Targets and Protection Profiles editor using predefined libraries for identifying CC functional and assurance requirements; however, it is not available online anymore. Finally, CC Toolbox was an MS Windows application sponsored by the National Information Assurance Partnership (*i.e.*, a US government initiative) to assist users in editing security targets but is no longer supported and available. (Version 6.0f, March 2001) [19].

So far, to the best of our knowledge no open-source solution has been developed and adapted to the needs of the connected vehicles security evaluation; furthermore, previous approaches that are not tailor-made for automotive usage can apply only on certain platforms, are typically restricted to certain evaluation tasks (*e.g.*, security target evaluation) or require already certified products. AFT has been designed to advance the relevant state-of-the-art by addressing those gaps.

III. The AFT design principles

This section presents the reference architecture of the AFT, realized as a modular software platform. First, it goes through a brief analysis of the involved technical requirements that relate to a broad set of needs. Essentially, the proposed architecture can support the implementation of a toolkit that would be used to assist CC-based evaluations supporting numerous evaluation classes. The derived architecture retains a certain level of generality designed to follow well-known software standards (such as HyperText Markup Language for web pages and cross-browser compatibility, ECMAScript which is the standard javascript specification for dynamic content [21] as well as the Cascading Style Sheets [22] to specify a standardized way for browsers to present content). This enables its quick adaptation to security evaluation needs and at the same time serves as the basis for the subsequent AFT software development.

AFT is enabled to host the identified automotive assets (*i.e.*, security controls to protect potentially confidential data are applied), the involved threats, assumptions and security objectives as well as the relevant SFRs (included in the ASE class [3]). It also provides the opportunity for the ToE developer to describe the functional specifications using a graphical tool (ADV class) and specify relevant test for each ToE interface (ATE class). Towards that end, the toolkit is expected to meet a set of technical requirements, typical in most complex software systems: Adaptability is mainly a result of the AFT modular design while auditability (secure logging), backup, capability for timely updates and testing are standard software requirements. Highly important requirements are the AFT's extensibility and interoperability needed to be able to cover the full spectrum of CC evaluation needs and expose its functionality to third parties, respectively. Data must be safely stored allowing for various level of confidentiality (*i.e.*, some automotive data are sensitive). A detailed documentation accompanies the published code of the toolkit that scales to support large number of users and data volumes.

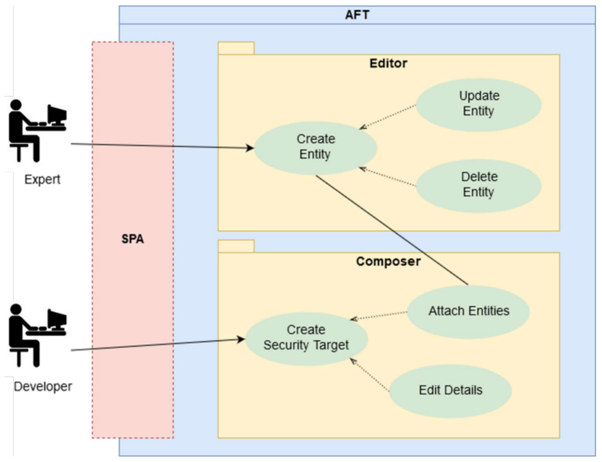


Fig. 2. AFT user-interfaces and basic functionality for the evaluation of the product's security target (ASE class)

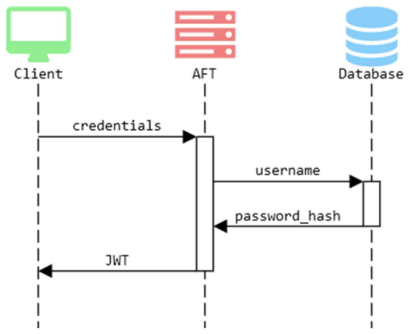


Fig. 3. The Single Page Application (SPA) mode of operation

There are two basic roles of users defined in the AFT architecture (see Fig 2): The 'expert' and the 'developer'. The former is the one who can define entities and relations in the toolkit; the latter is the main (typical) audience of AFT *i.e.*, the automotive product (ToE) developer who will rely on the AFT operation to efficiently compile the CC evaluation inputs (to be provided to the evaluator).

A. *The AFT front end design*

The aforementioned users may interact with the Toolkit relying on one interface, the Single Page Application (Fig. 2). This provides a unified yet modular access to the application, offering a number of benefits: There are distinct client and server components. The Toolkit then will have a clear separation between user interface and business logic ensuring modularity (related to the requirements of extensibility and scalability). Furthermore, all communication between client and server code occurs with AJAX calls implying that AFT will have a ready Machine-to-Machine interface (*i.e.*, adaptability) without any extra effort. A single-page application (Fig 3) works by first delivering the web page along with all functionality in the initial browser request. After that initial load the page updates itself leveraging small asynchronous requests to the web server (which improves performance).

B. *The AFT back end design*

Dot Net Core is a multi-platform open-source software framework [9]. It contains a compiler and a runtime environment as well as a package manager for developing and running C#, F# and VB applications. It supports console and web applications but not native user interfaces. The platform comes with built-in tooling to support the development process.

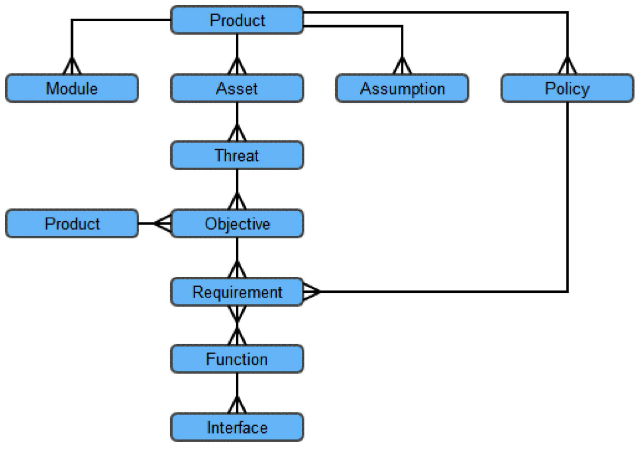


Fig. 4. AFT schema for the Security Target compilation

Specifically, for web applications the platform provides the ASP.NET Framework. It can be used both for web user-interface (UI) and web API modular applications with a focus on clean design and testability (meeting the relevant AFT requirement). It interfaces well with all single-page application frameworks due to the Model-View-Controller component which is complete and well-implemented. Another part of the .NET stack is the Entity Framework (EF) Core, an object-relational mapping library. EF Core allows for both mappings on an existing database as well as table generation from available models. The latter approach allows for easier integration into applications (*i.e.*, adaptability) and also promotes testing (testability requirement).

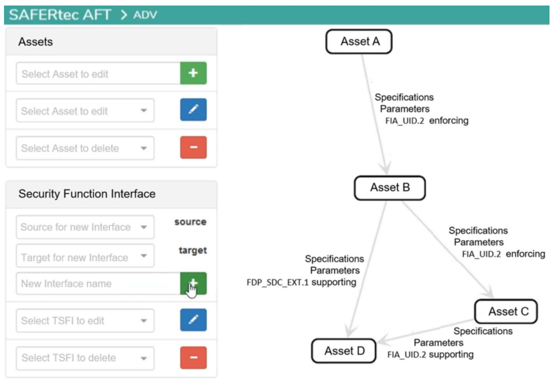


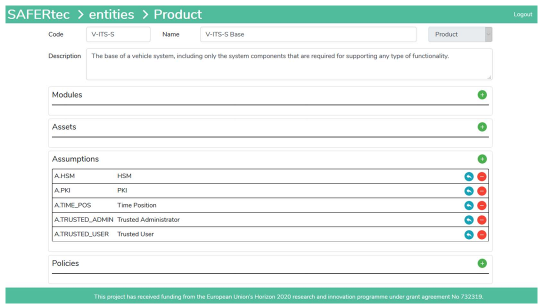
Fig. 5. AFT graphical tool to support the Common Criteria ADV evaluation

C. *Basic functionality*

For its basic functionality AFT realizes a data modelling structure comprised by a set of entities and their relations (*i.e.*, EF Core, mentioned above). Those entities seek to accurately represent the building blocks (or sequence of mandatory inputs) for the ST compilation. Fig 4 represents the conceptual draft of the entities which resides in the database and is exposed for editing to the users. The core object is the Product (*i.e.*, the ToE) around which all others are centred with most of the relations being of 'one-to-many' type. This E-R model and the relevant AFT implementation assists the user in defining the ST for the ToE. Further support is provided with a graphical tool (see Fig. 5) that allows the accurate description of the ToE's design and interface specifications covering tasks of the ADV class. This is done using a diagramming client-side library. Finally, relevant tests can be selected for each ToE interface from a pre-defined list (ATE class). The AFT architecture and interfaces have been carefully designed to 'force' the ToE developer to provide the CC mandatory elements and justify all associated relationships.

IV. The AFT Implementation

To meet the design specifications of Section III, the AFT should support two main functionalities. The first one is an editor for entities. The developer can create and modify entities belonging to the categories specified by the CC methodology. The addition or removal of relationships between these entities


SAFERtec > entities > Product

Logout

Code	V-ITS-S	Name	V-ITS-S Base	Product

Description: The base of a vehicle system, including only the system components that are required for supporting any type of functionality.

Modules

Assets

Assumptions

A.HSM	HSM
A.PKI	PKI
A.TIME_POS	Time Position
A.TRUSTED_ADMIN	Trusted Administrator
A.TRUSTED_USER	Trusted User

Policies

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732319.

Fig. 6. Adding a set of assumptions (taken from our modular protection profile [2]) to an automotive target of evaluation (ToE)

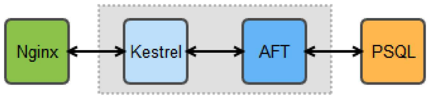


Fig. 7. The AFT deployment chain

is supported. The end-goal is to realize a structure holding all relevant data of one or more Protection Profiles (PPs). The second functionality is where the developer can leverage the previous information to author the Security Target (ST). The ST editor contains a set of sections guiding the ToE developer in selecting those CC inputs which apply to the product. Additionally, the developer can add relevant entities which will only be used by their Security Target (see Fig. 6).

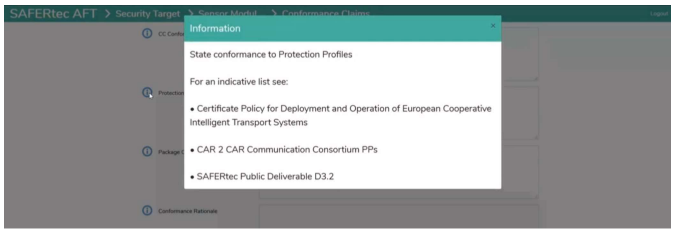
A. *The AFT front end implementation*

The front end was implemented in TypeScript and uses the Angular SPA framework (Fig. 3). This allows for a modular and composable architecture with reusable elements. The basic types of units are the user-interface building blocks and services constituting the main task-performing classes. The entity editor contains a dynamic list of relations to other entities which changes according to their type; the user can efficiently navigate through them. The AFT client side communicates with the back end via the REST interface minimizing contrary to more traditional web paradigms, the data exchanged during operations.

B. *The AFT back end and dedicated features*

The server component is the more logic-heavy part of the application, written in C# and run on the .Net Core framework [9]. It contains the controllers which comprise the REST interface of the application and another set of services which are responsible for most functionality. The application along with the kestrel web server are the main executables (see Fig. 7). The server sits behind a reverse proxy and also communicates with a PostgreSQL database.

The proposed AFT implementation retains a certain level of generality; it realizes the functionality that a CC security evaluation requires regardless the application domain (see, for instance, the implemented schema in Fig. 4). Then, to tailor AFT for the needs of connected vehicles evaluations we have a) populated it with relevant automotive data and b) developed dedicated supportive functionality. We detail both aspects in the following points:


SAFERtec AFT > Security Target > Sensor Modul > Conformance Claims
Logout
CC Confor
Protection
Package C
Conformance Rationale
×

Information

State conformance to Protection Profiles

For an indicative list see:

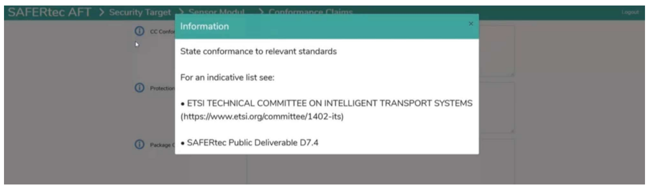
• Certificate Policy for Deployment and Operation of European Cooperative Intelligent Transport Systems

• CAR 2 CAR Communication Consortium PPs

• SAFERtec Public Deliverable D3.2

Fig. 8. AFT-provided pointers and technical notes on Protection Profile conformance claims

- AFT has been populated with vehicular data (*i.e.*, threats, security objectives, functional requirements *etc*. for certain ToEs) provided by our connected vehicles modular Protection Profile [2] as well as the reference TVRA report (*i.e.*, Threat, Vulnerability and Risk Analysis for Intelligent Transport Systems) published by the European Telecommunications Standards Institute (ETSI) [24]. The user can select the appropriate data for the ToE under evaluation or add new.
- AFT data has been shaped according to the experimentation with our real-world V2I testbed [11]. A broad set of functional requirements falling under V2I communication instances have been tested and the results have been incorporated into the AFT database.
- Special functionality has been added to guide the automotive product developer in the demanding compilation of CC evaluation inputs (which without the assistant of tools incurs increased costs, see Section V). AFT provides pointers to external technical documents, relevant standards and templates that appropriately guide the user to correctly fill-in the needed CC inputs (for instance, state the conformance to PPs and standards in Fig. 8 and Fig. 9, respectively) and therefore, considerably eases the evaluation process.
- Last but not least, the ability to easily update the AFT implementation (in terms of the maintained automotive database) can -to some extent- address the security evaluation of highly-frequent vehicular software updates.


SAFERtec AFT > Security Target > Sensor Modul > Conformance Claims
Logout
CC Confor
Protection
Package C
×

Information

State conformance to relevant standards

For an indicative list see:

• ETSI TECHNICAL COMMITTEE ON INTELLIGENT TRANSPORT SYSTEMS (https://www.etsi.org/committee/1402-its)

• SAFERtec Public Deliverable D7.4

Fig. 9. AFT-provided links to online documents and technical notes on conformance to automotive standards

V. Empirical Cost Evaluation and Implications

In this section we first provide an empirical cost analysis of the AFT usage and then discuss the 'application domain' and potential implications of the toolkit.

Comparing the regular *i.e.*, unassisted by tools, CC evaluation cost with the one incurred by AFT is not trivial. It would require numerous CC-based evaluations with and without AFT to provide actual figures. This calls for considerable funds and takes years to conclude. Thus, the *only* approach that can currently offer some evaluation insights is empirical. What we aim to do here in view of the AFT code release, is to rely on our long experience (*i.e.*, our numerous CC-based product evaluations revealing what the involved actors possess as expertise and their capability to prepare input documents[2])

[2] For certain evaluation families, many standardized processes used by OEMs (*i.e.*, car manufacturers) and Tier-1 suppliers already require the generation of similar documents (*e.g.*, quality process requirement of safety management enforced by standards such as ISO 26262)

TABLE I
REGULAR *vs.* AFT-BASED EVALUATIONS: EMPIRICAL COMPARISON OF REQUIRED EFFORT AND TIME FOR VARIOUS ASSURANCE CLASSES

Assurance component	**Task Input**	**Regular (*i.e.*, unassisted) CC efforts**	**Using the AFT**
ASE	Security Target Document	**ST writing:** 3 days to instantiate a PP Extra efforts per evaluation task cycle: 0,5 days **ST evaluation:** 2 days Extra efforts per evaluation task cycle: 0,5 days	**ST writing:** 2 days to instantiate the AFT PP Extra efforts per evaluation task cycle: 0,5 days **ST evaluation:** 1 day Extra efforts per evaluation task cycle: 0,5 days
ADV (ADV_FSP)	Documents describing the ToE interfaces and association with SFRs Error summaries for each ToE interface	**Initial documentation**: 7 days Extra evaluation efforts: 2 days Extra efforts per evaluation task cycle: 0,5 days **Documents evaluation** Iteration 1: 2 days Iteration 2: 2 days Higher iterations: 0.5 days **Cost estimation**: 3 working days	Estimated time needed for extra evaluation to be *reduced up to 50%* as AFT provides graphical tools & templates Better quality of inputs (more structured and harmonized) to *reduce evaluation time up to 30%*
ATE (ATE_FUN)	Test plan Test results	**Document production** (ToE developer) Initial documentation: 4 days Extra efforts for evaluation: 2 days Extra efforts per evaluation task cycle: 2 days **Document evaluation** Iteration 1: 3 days Iteration 2: 2 days Higher iterations: 0,5 days **Cost estimation**: 5 working days	AFT provides the functionality to associate the main vehicular interfaces with a set of proposed tests for the ATE needs Estimated time for initial documentation and extra efforts for the evaluation to be *reduced up to 50%* thanks to AFT templates

and empirically assess the expected AFT gain in effort and time (see Table I). We focus on three CC evaluation classes currently covered by AFT and discuss the effort in working days needed for both the production of the input and the task evaluation (corresponding to medium assurance level) with and without the usage of AFT. Working days are associated with specific expertise profiles for which we provide estimated average person/day values; those imply funds, also needed to cover the high cost of the evaluator. Clearly, only *high-level reasoning* can support the way we derive our estimations, considered as averages across the evaluation of different ToEs (of various sizes and complexity) requiring 30-40 inputs documents [3].

A. Security target (ST) evaluation (ASE)

AFT already relies on a dedicated modular Protection Profile (PP) [2] for connected vehicles which eases both the editing and the evaluation of the ST. In Table I we present the corresponding efforts for this evaluation task. It is again important to note that the reported values constitute empirical estimations based on our expertise since no public studies exist on the matter and the results from the actual AFT usage are yet to come. However, PP instantiation is common in CC evaluations and the relevant difficulties and time-consuming activities are well known (*e.g.*, the ToE developer cannot properly identify the type/precision of input data or fails to specify SFRs due to lack of examples etc.). Furthermore, examples and technical notes describing what is expected in each step have been foreseen in AFT to provide guidance (see Fig. 8). In general, our estimation suggests a gain of 1 day over average PP instances. The main rationale behind that is the reliance on the dedicated PP accounting for automotive standardized elements and architectures thus, limiting the possible SFRs that should be otherwise thoroughly considered.

B. Architectural design evaluation (ADV)

The functional descriptions mainly help complete the full tracing proof to move from the SFRs (defined in the ToE's security target) to the product function and its interfaces, implementing the SFRs to be evaluated. The functional description is at the interface-level and must provide detailed information including the protocols used over those interfaces. For each interface, the security functions accessible through-it should be provided. For each security function, the ToE developer needs to provide a broad set of inputs such as its purpose, the relevant SFR it enforces (extracted from the security target), the interface and exchanged data, the error messages as well as security configuration-parameters. Such descriptions must be carefully edited to correspond to the ToE functionality description, detailed in the security target document. Consequently, all required information is to be formatted under the CC structure rather than a regular product description and many links (*i.e.*, tracing) to the security target elements should not be forgotten. Thus, AFT is expected to considerably favor the ADV class (see Table I). It helps the developer automatically identify all mandatory content and ensure that all necessary elements/justifications have been included. Without tools, this task becomes much more difficult and takes longer to achieve.

C. Functional and independent tests evaluation (ATE)

The ToE developer should provide a test plan including test scenarios or test scripts. Detailed information for each scenario such as prerequisites, operations, and expected results are also needed. The provided documents and results should justify why tests are sufficient to cover every interface related to ToE security function, earlier identified along the ADV class; an automated verification of this coverage and the associated rationale is clearly easier and faster. Moreover, the AFT structured information (*i.e.*, vehicle assets, interfaces, SFRs) can serve as a basis for the test plan definition. AFT provides the needed association of the main automotive interfaces with a set of proposed tests (most of which have been earlier examined in our testbed [11] showcasing their relevance). Thus, increased savings for the ToE developer are expected due to that guidance (3rd entry in Table I) and also for the evaluator who should now receive higher-quality evaluation documents.

D. The AFT implications

The implications of the introduced toolkit are broad. AFT, when fully populated with all needed data (*e.g.*, road-side unit Protection Profile data which the literature currently lacks), can be used by security evaluators as well as benefit a large part of the automotive industry supply chain such as OEMs and Tier-1 suppliers. The generated Security Targets of various involved products (*e.g.*, on-board units, *etc.*) already include details (*e.g.*, SFRs) that are relevant for security engineers of the automotive industry. Importantly, the more the available data the higher the AFT efficiency. Finally, an interesting implication points-to the AFT's standardization potential; as evaluation input data need to respect the strict CC structure it is not easy for an automotive product developer to know if the information he provides is in the appropriate structure/format. AFT, if standardized, would require inputs that meet both CC and automotive technologies constraints. That would help automotive developers produce CC-compliant evaluation inputs, ease the evaluators and further lower the involved costs.

VI. Conclusions

As the cyber-physical system of connected vehicles increasingly relies on data exchanges (with the infrastructure), further cyber-security concerns are raised requiring high assurance levels about their fulfillment. Only the most credible assurance framework *i.e.*, the Common Criteria (CC) standard can provide such assurance but comes with high resource and time costs. To minimize them, we have designed, implemented and introduced a platform-independent open-source online toolkit (AFT) that the state-of-the-art lacks. Drawing on dedicated automotive requirements (Protection Profile) and exploiting our previous test-bed experimentation results, we have populated AFT with automotive data and appropriate technical notes to enable the efficient compilation of evaluation entries for three demanding CC classes. As the actual AFT performance figures would come after years of (products) security evaluations, we resort to an empirical yet fairly justified estimation suggesting that for some cases the AFT usage-gains could even be more than 30% compared to unassisted evaluations. The toolkit (of significant extendability features) is offered to automotive industry experts (from OEMs) and security assurance evaluators aiming to increase trust in connected vehicles.

References

[1] P. Pantazopoulos *et al.*"Towards a Security Assurance Framework for Connected Vehicles," in Fifth IEEE Workshop on Smart Vehicles: Connectivity Technologies and ITS Applications, Chania, Greece, 2018.

[2] K. Maliatsos *et al.*"Standardizing Security Evaluation Criteria for Connected Vehicles: A Modular Protection Profile", IEEE Conference on Standards for Communications and Networking", Granada, Spain, 2019.

[3] "ISO/IEC 15408 part 1/2/3:2005-Information technology, Security techniques, Evaluation criteria for IT security," Tech. Rep., v3.1, Release 5. [Online]. Available: https://www.commoncriteriaportal.org/cc/

[4] N. Lu *et al.*"Connected vehicles: Solutions and challenges." IEEE Internet of Things Journal, vol. 1, no. 4, pp. 289-299, Aug. 2014.

[5] H. H. Thompson, J. A. Whittaker, and F. E. Mottay, "Software security vulnerability testing in hostile environments," in Proc. of the ACM Symposium on Applied Computing, NY, USA, 2002, pp. 260–264.

[6] National Institute of Standards and Technology, FIPS PUB 140 Series: Requirements and standards for cryptographic modules, May 2001. [Online]. Available: https://csrc.nist.gov/publications/fips

[7] United States Department of Defense, "Trusted Computer System Evaluation Criteria (Orange Book)," Tech. Rep., 1985.

[8] C. Jahl, "The information technology security evaluation criteria," in The 13th Int'l Conf. on Software Engineering, May 1991, pp. 306–312.

[9] [Online]. Available: https://dotnet.microsoft.com/download

[10] E. Schoitsch *et al.*"The need for safety and cyber-security co-engineering and standardization for highly automated automotive vehicles," in Advanced Microsystems for Automotive Applications 2015. Springer Publishing, 2016, pp. 251–261.

[11] A. Marchetto *et al.* "CVS: Design, Implementation, Validation and Implications of a Real-world V2I Prototype Testbed", 91st IEEE Vehicular Technology Conference: VTC-Spring, Antwerp, Belgium, May 2020.

[12] M. Hashem Eiza and Q. Ni, "Driving with sharks: Rethinking connected vehicles with vehicle cybersecurity," IEEE Vehicular Technology Magazine, vol. 12, no. 2, pp. 45–51, June 2017.

[13] IEEE Conformity Assessment Program (ICAP). [Online]. Available: http://standards.ieee.org/about/icap/index.html

[14] S. Haddad *et al.*"CARSEM: A Cooperative Autonomous Road-vehicles Security Evaluation", 25th ITS World Congress, Copenhagen, Denmark, 17-21 September 2018.

[15] Common Criteria Recognition Arrangement (CCRA) members [Online]. Available: https://www.commoncriteriaportal.org/ccra/members/

[16] Lightship Security Common Criteria accredited laboratory e-book [Online]. Available: https://lightshipsec.com/download/Lightship-7-Steps-to-Common-Criteria-eBook.pdf

[17] D. Rogowski, "Software Implementation of Common Criteria Related Design Patterns," 2013 Federated Conference on Computer Science and Information Systems, Krakow, 2013, pp. 1147-1152.

[18] D. Horie *et al.* "GEST: A generator of ISO/IEC15408 Security Target templates", Computer and Information Science 2009, pp 149-158.

[19] CC Toolbox [Online]. Available: http://pagenotes.com/writings/ccToolbox6f/

[20] Trusted Labs Technologies [Online]. Available: https://www.trusted-labs.com/

[21] ECMAScript Language Specification. [Online]. Available: https://www.ecma-international.org/ecma-262/

[22] W3C Cascading Style Sheets: https://www.w3.org/Style/CSS/Overview.en.html

[23] EU SAFERtec project: Security Assurance Framework for Networked Vehicular Technology (2017-2020), [Online]. Available: "https://www.safertec-project.eu"

[24] ETSI TR 102 893. 2017. Intelligent Transport Systems Security, Threat, Vulnerability & Risk Analysis (TVRA) V1.2.1. [Online]. Available: https://www.etsi.org/