Generation of Cyberattacks Leading to Safety Top Event Using AltaRica: an Automotive Case Study

Théo Serru, Nga Nguyen, Michel Batteux, Antoine Rauzy, Raphael Blaize, Laurent Sagaspe, Emmanuel Arbaretier

Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

Generation of Cyberattacks Leading to Safety Top Event Using AltaRica: an Automotive Case Study

Serru Théo
ETIS UMR 8051
CY Cergy Paris University, ENSEA, CNRS
Cergy, France
Airbus Protect, France
theo.serru@ensea.fr

Nguyen Nga
Leonard de Vinci Pole Universitaire
Research Center
Paris La Defense, France
nga.nguyen@devinci.fr

Batteux Michel
IRT SystemX
Palaiseau, France
michel.batteux@irt-systemx.fr

Rauzy Antoine
NTNU
Trondheim, Norway
antoine.rauzy@ntnu.no

Blaize Raphaël
Airbus Protect
Blagnac, France
raphael.blaize@airbus.com

Sagaspe Laurent
Airbus Protect
Bordeau, France
laurent.sagaspe@apsys-airbus.com

Arbaretier Emmanuel
Airbus Protect
Elancourt, France
emmanuel.arbaretier@apsys-airbus.com

***Résumé*—Les systèmes cyber-physiques critiques sont à risque pour leurs utilisateurs et leur environnement. Il est donc indispensable d'analyser tous les risques liés à leur utilisation, y compris les risques liés aux cyberattaques. Ainsi, dans cet article, nous présentons une méthode permettant d'analyser les risques de cyberattaques sur les systèmes cyber-physiques, et leurs effets sur la sécurité (safety). Nous utilisons le langage AltaRica et l'outil SimfiaNeo pour modéliser un cas d'étude automobile et son comportement en cas d'attaque. Nous montrons également comment générer les séquences d'événements menant à une situation redoutée. Cependant, les modèles basés sur les systèmes à événements discrets, comme les modèles AltaRica, sont sujets à l'explosion du nombre d'états. Afin de contrer ce problème, nous présentons un nouveau filtrage permettant de réduire le temps de calcul et de ne garder que les séquences d'attaques les plus pertinentes.**

***Abstract*—Cyber-Physical Systems (CPS) are safety critical as they can be at risk for their users and environment. It is thus paramount to assess all risks related to their use, including the risks of cyberattacks. Indeed, malicious intruders are more and more active on such systems because of their interconnected nature. In this paper, we present a method to assess the risks of cyberattacks on CPS and their effects on safety. Thus we will show how to use the language AltaRica and the tool SimfiaNeo to model the behavior of these systems in case of cyberattacks and to generate the sequences of events leading to a safety critical state. Models based on discrete event systems like AltaRica are, however, subject to state-space explosion. To overcome this issue, we will present a new cutoff, called footprint, used to reduce the computation time and keep only relevant sequences of events. To illustrate the effectiveness of our work, we will use an automotive case study as an illustration.**

***Keywords*—Cyberattacks, Safety, Sequences of Events, AltaRica, SimfiaNeo.**

This research project is funded by CY Initiative d'Excellence and Airbus Protect.

I. Introduction

Cyber-Physical Systems (CPSs) are intensively used in the industry. Due to their highly interconnected nature, they need to be carefully designed and protected. Model-based analyses provide a powerful framework to model the behavior of these systems. Indeed, the modeling of CPS and their behavior in the presence of faults has benefited from the high expressiveness of Discrete Event Systems (DES), such as stochastic Petri nets [1], Figaro [2] or AltaRica [3]. One of their major benefits is the generation of scenarios of events (initially failures) leading to a safety critical state. From the sequences, it is then possible to elaborate an effective risk mitigation strategy and to improve the architecture of the system.

With CPS, the increasing number of embedded software raises additional risks related to cyberattacks. Therefore, it is paramount to assess the risks of cyberattacks and their impacts on safety [4]. In this article, we show how to use AltaRica for the analysis of cyberattacks on CPS and how to generate sequences of attacks leading to a safety critical situation.

An issue arising from the exploration of systems exposed to malicious intruders is the state-space explosion. Thus, to reduce the explosion, we introduce a novel criterion called footprint. This criterion takes advantage of the dependent nature of cybersecurity events to filter the sequences of events.

Therefore, the contributions of this article are twofold. First, we introduce a new cutoff, called footprint, to reduce the state-space explosion while exploring sequences of cyberattacks. Second, we evaluate the relevance of this new criterion and compare it to the cutoffs traditionally used in Model-Based Safety Assessments (MBSA). Finally, the contributions are

illustrated thanks to the case study of an automotive, modeled in AltaRica Data-Flow.

The reminder of this article is organized as follows. First, Section II introduces the context of formal modeling of cybersecurity properties, as well as existing works on this topic. Section III depicts the specificities of the automotive case study. Then, Section IV shows how to model the case study with AltaRica and how to generate sequences of events leading to a safety critical state. Section V analyzes the sequence generated and introduces the new cutoff for cyberattack sequences. Finally, Section VI concludes this work and addresses future perspectives.

II. Context

A. Model-Based Analyses in Cybersecurity

Model-based analyses rely on modeling for the analysis of systems. Cyber security refers to the body of technologies, processes, and practices designed to protect networks, devices, programs, and data from attack, damage, or unauthorized access. Among such practices, we may find cryptography, penetration testing, code analyses, vulnerability risk analysis, etc. In this work, we will use a model-based approach to automatize some of the cybersecurity engineers' tasks. Thus, we quickly define how models can be applied in cybersecurity and why they cannot be used for all the cybersecurity processes. First, activities such as penetration testing and code analyses are not likely to use model-based methodologies. To find a new vulnerability in a code, an analyst need to analyze the code itself without abstraction because vulnerabilities hide in plaintext, waiting to be discovered. The analyst has thus several tools to study the program, but cannot use an abstraction as it hides the lines of code behind a model. The attacks used can be automatized, but the target must be the real system (or part of it).

Then, model-based security comes after the discovery of vulnerabilities, to analyze their consequences and strengthen the system. It allows to work at a higher abstraction level than the code level. Some model-based approaches focus on automated formal verification of code [5], [6], security protocols [7], [8] or cybersecurity properties (such as confidentiality, availability, integrity, non-repudiation, etc.) [9], [10], [11]. This last category is commonly called "Model-Based Security" (MBS) or "Model-Based Security Engineering" (MBSE) and is the focus of our work. It focuses on verifying security properties such as confidentiality, availability, integrity, non-repudiation, etc. against cyberattacks and at modeling the effects of cyberattacks on these properties. In addition, MBSE is more and more used to assess CPS security properties as surveyed in [12], [13]. The main difficulty in CPS cybersecurity is linked to the numerous interactions between the physical and cyber components.

In this work, we thus use models in an assessment suitable to the use of models. This is the risk assessment of CPS with a focus on safety. Here, we aim to evaluate the effects of cyberattacks on safety, to avoid catastrophic events that could result from a cyberattack. To do so, we abstract the behavior of CPS with DES and the AltaRica language and represent how it reacts when cybersecurity-related events occur. Therefore, we will not talk about "model-based security" when referring to our work, but "analysis of the impact of cyberattacks on the safety of CPS" to avoid any confusion for the reader.

Finally, we insist on the need to abstract the behavior of the system as it is intractable to model the functioning of every software embedded in a CPS. In this work, we use a system abstraction level, inherited from safety assessments, to keep a reasonable complexity and a sufficient expressiveness.

In the next subsection, we present works related to the modeling of cybersecurity properties and their impact on the safety of systems.

B. Related Works

A wide variety of works have been focusing on modeling CPS and extracting critical sequences, and we present some of them in this section.

First, a very common formalism is attack trees (AT), [14], [15]which has been widely used to assess cybersecurity risks with the possibility to generate cut-sets. Apart from classical AT, works like [16], [17], [18] have extended them to consider dynamic behaviors, countermeasures or ordered actions. Other works use timed automata [19] or Markov chains [20], [21] to use time or cost constants.

Then, attack graphs are used in works such as [22], [23] to generate attack paths and propose mitigation or detection mechanisms. Another work [24] proposes to use attack graphs in the reliability assessment of a SCADA system. [25] uses graphs and an "event model" to generate attack paths. Such event model is also used in [26], [27] where pre-conditions and post-conditions of attacks are used to automatically build sequences. Others [28], [29], [30] aim to generate attack paths with a focus on the attacker's behavior and capabilities, but without representing the system's architecture.

Finally, other works such as [31], [32], [33] use formalisms allowing to represent the architecture of the system and consider safety properties.

In the above presented works, the main limitation is the state space explosion. Indeed, when modeling CPS and their behavior, many attacks or components have to be represented. In addition, we believe that modeling the architecture of the systems is of critical importance as it eases the understanding of the model and helps to identify how to improve the architecture. Then, in the remaining of this article, we will present a model that can solve these issues.

III. Case Study

A. Introduction

This work is illustrated by an automotive case study taken from the European project EVITA (E-safety vehicle intrusion protected applications) [34]. From 2008 to 2011, members of this project have worked on a unified approach to security and safety risk analysis for automotive on-board networks. In addition, they have developed a relevant use case and several security threat scenarios, which are available online. Therefore,

we view this project as an opportunity to use a case study widely accessible and to challenge our model, even if it is not up to date.

B. Architecture

The reference architecture is depicted in Figure 1, taken from the deliverable D2.1 of the project [35].

This architecture consists mostly of sensors, actuators and ECUs. It aims to represent classical on-board network architecture, clustered in different domains for Communication, Powertrain, Chassis & Safety, Body Electronics and Infotainment (Head Unit). Finally, the interfaces of the system are the communication unit (CU) and mobile devices connecting to the head unit (HU).

The CU is equipped with vehicle-to-vehicle and vehicle-to-infrastructure communication. The vehicle can thus receive (and send) brake notification from (to) a car in the neighborhood. After the reception of this message, the vehicle compares its position with the sender and eventually activates the brake function.

C. Dark-Side Scenarios

To identify potential security threats, we refer to the deliverable D2.3 [36] of the project. In this document, the authors developed "dark-side scenarios" and modeled them as attack trees [15]. In details, they identified several attack goals and constructed attack trees that could achieve the goals. The trees are based on the functionalities of the system identified in the deliverable 2.1 [35]. Then, among the proposed attack goals, we choose two that impact the passengers' and system's safety.

a) Unauthorized brake: The first unwanted event is the activation of the brake function by the attacker. It can result from an attack on the environment sensors, as well as a manipulation of the Chassis Safety Controller (CSC). Another threat would be a fake brake notification from another car.

b) Attack active brake function: In the second unwanted event, we consider an adversarial trying to attack an authorized braking event. To do so, it is possible to delay, inhibit, or degrade the quality of the brake. Once again, an attack on the environmental sensors is possible as well as a Denial of Service (DoS) on the CU or CSC.

D. Threats

Our starting point is to represent the threats modeled in the leaves of the trees from [36]. We detailed these threats in Table I.

In this work, we wish to go deeper in analyzing the risks of cyberattacks. The threats presented in [36] are inspired by the functioning of the system introduced in [35] but the trees don't specify how the attacker penetrates the system. Furthermore, there is no notion of attack progression within the system. We believe such information is valuable for safety and security engineers and would allow an effective mitigation strategy. Therefore, in Section IV-A, we will model how the attacker can penetrate the system, propagate, and reach its goal. Thanks to the properties of DES, we will also be able to model the dynamic behavior of the system and its countermeasures.

Table I
SOME THREATS CONSIDERED IN THE CASE STUDY, INSPIRED FROM [36]

Component	Threats
Communication Unit	DoS, Data injection, Flash or forward malicious code encrypted as an update, Data injection
Digital Short Range Communication (DSRC)	Jamming, Data transmission delayed
Universal Mobile Telecommunications System (UMTS)	Download malicious code encrypted as an update, Flooding
Environmental Sensors	Manipulate sensors information or environment, Flash malicious code as an update
Body Electronic Module, Powertrain Controller	Data injection, Flash malicious code as an update, Send false brake notification
Chassis Sensors	DoS, Flash malicious code as an update
Chassis Safety Controller	DoS, Flash or forward malicious code as an update, Exploit protocol parsing flaws or state handling
GPS	Spoofing
Head Unit	Flash malware as an update, Data injection, Exploit traffic message vulnerabilities, Send false brake notification
Brake Controller	Automatic brake prevented or degraded, Unwanted brake activation

E. Countermeasures

We slightly changed the architecture of the system to add two security measures. Their goal is to make it more difficult for an attacker to penetrate or corrupt the system and to evaluate how it will influence the sequence generation. The first countermeasure is a redundancy of the *environment sensor*. With this redundancy, the attacker will have to manipulate both sensors to trigger the unwanted situation. Then, the second countermeasure is an integrity module between the *UMTS* and the *CU*. Such module will detect malicious data being sent from one component to the other. To send malicious data, the intruder will have to shutdown the integrity module, or stay undetected.

Finally, we only consider remote attacks on the system as it is very unlikely that an attacker crashes the car with the risk of being injured.

IV. MODELING AND ANALYSIS OF THE CASE STUDY

A. Modeling with AltaRica

The automotive case study introduced in the previous section is modeled using AltaRica Data-Flow [37]. This language is used in industry to model systems along with their safety properties. AltaRica Data-Flow can be seen as a generalization of both Petri nets and Block Diagrams [38]. To Petri nets, it borrows the notion of states, events, and guarded transitions.

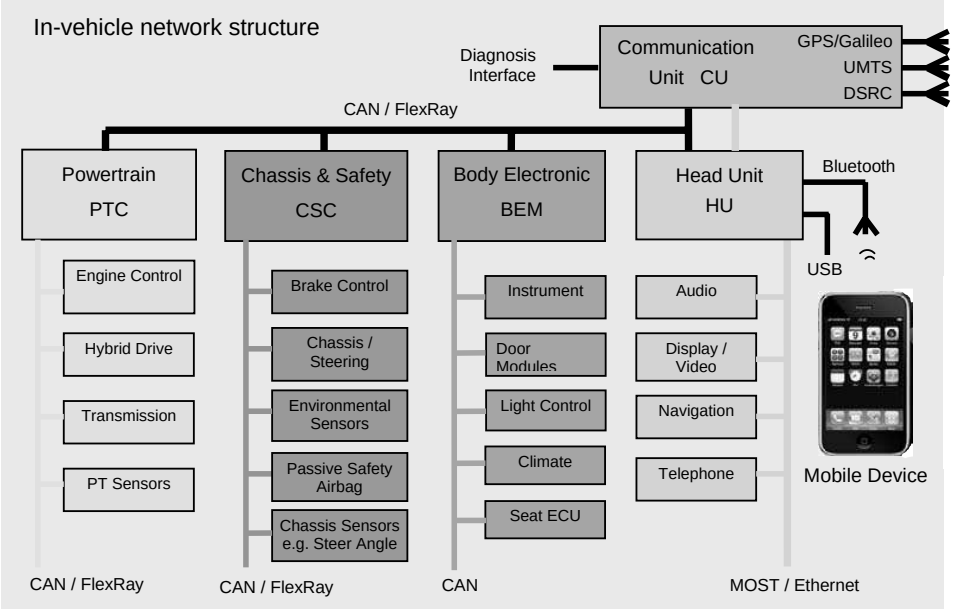

Figure 1. EVITA Automotive Reference Architecture [36]

To Block Diagrams, it borrows the notion of hierarchical descriptions and flows circulating through a network. The tool SimfiaNeo used in this work has been developed by Airbus Protect and allows us to model and analyze the safety of systems. It is built on the language AltaRica Data-Flow and thus benefit from its expressiveness.

The model of the system is based on the architecture depicted in Figure 1. From this architecture, we took off the components that are not involved in the scenarios. The remaining includes 14 components modeled as AltaRica nodes. Full code of the AltaRica model is available on demand. Contact the authors if interested.

In a component, we model the actions of the attacker and system reconfiguration as *events*. The transition (*trans*) is the mechanism used to actualize the value of *state variables*, where *state variables* represent properties of the components. A transition is written as such: *Boolean formula* $|-$ *event's name* $\rightarrow$ *instruction*. For example, in in the following transition:
$(role = root)\ and\ (data\_out = nominal)\ |- Manipulate\_sensor\_information \rightarrow data\_out := erroneous$;
manipulating environmental sensors' information is possible for the attacker if the attacker has the root privilege and if it hasn't been done before (i.e. output data is nominal). The result of such action is the output data being erroneous. Finally, assertions (*assert*) are used to model flux propagation. It is used to propagate the value of *flux variables* from a component to another.

The resulting AltaRica model contains 14 components, 21 links and 60 transitions. Figure 2 shows the graphical model of the case study as printed in SimfiaNeo. The transitions are based on the threats identified in III-D. As specified in II-A, the model does not allow to identify new ones but explores all combinations from existing threats. In the following, we present how to generate the scenarios resulting from this combinatorial.

B. Sequences Generated from the AltaRica Model

In this subsection, we use the sequence generator from SimfiaNeo to perform a qualitative analysis of the model. This tool runs through and generates the executions according to a target property and the constraints of the model. We thus have two target properties defined in Section III-C as critical states. In AltaRica, the properties are defined as observers using a Boolean formula. The first one relates to the unwanted activation of the brake function and is defined as: Brake_Controller.brake_activation=true (the variable "brake_activation" of the component "Brake_Controller" has the value "true"). The second observer relating to the attack on the "active brake function" is defined as: (Brake_Controller.integrity=partial_loss) or (Brake_Controller.availability!=nominal).

Then, we generate the sequences leading to these situations with a maximum length of 10 events. It resulted in 133429 sequences for the brake activation and 401625 for the attack on brake function. The huge amount of sequences generated is not surprising but prevents from analyzing them. This is because of the state-space explosion unavoidable in combinatorial models. Therefore, we can use two solutions to limit this explosion, minimality and cutoffs. Although these solutions are used in safety, they are relevant in a cyberattack point of view as they allow to measure the effort of the attacker and to filter the sequences based on this criterion. Obviously, the attacker (regardless of its experience) will try to reduce the

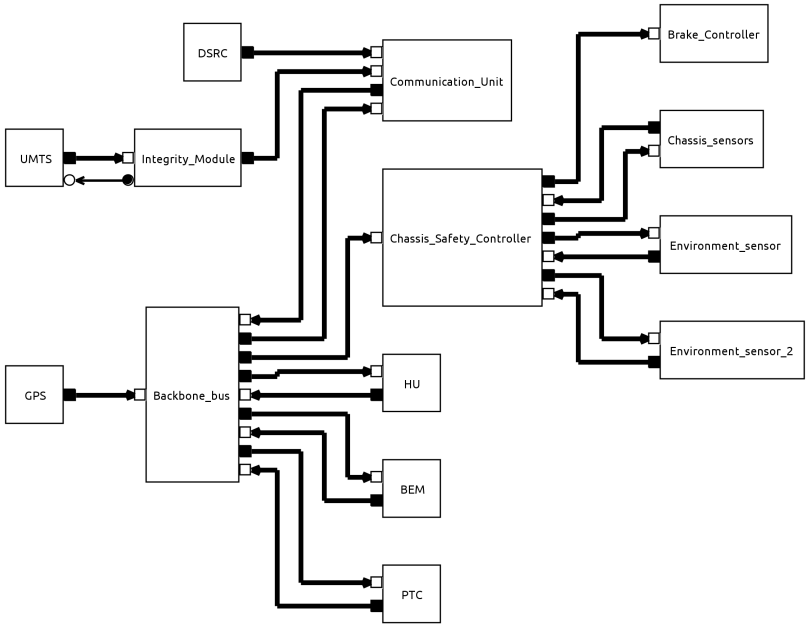

Figure 2. EVITA Automotive Graphical Model Made on SimfiaNeo

effort needed to achieve a goal and is likely to abandon if the effort required is too high. The following criteria (and the one of Section V) are thus chosen because they can filter sequences based on the effort of the attacker.

Minimal sequences are free of events having no effect on reaching the critical state. They are widely applied in safety assessment as they allow reducing effectively the number of sequences printed. SimfiaNeo generates 45 minimal critical sequences (MCS) leading to the unwanted brake and 116 leading to the attack on active brake function. A sequence is shown in Figure 3 for the two top events.

Some sequences generated, especially for the unwanted brake, can still be filtered. This would be a great deal, as it is difficult to manually analyze 116 sequences of length 6 to 10 events. Therefore, minimality is traditionally used along with cutoffs to obtain the minimal set of relevant sequences. Cutoffs used in MBSA are used to lower the state-space by taking advantage of the independent nature of accidental events. However, cyberattacks are mostly dependent and the traditional cutoff on the number of events in a sequence, called the order of the sequence, is not relevant. Therefore, the next section introduces a new cutoff dedicated to the generation of cyberattack sequences.

V. State-Space Reduction for Sequence Exploration

A. Introducing the Footprint

In this subsection, we introduce our mean to reduce the state-space explosion with sequences of cyberattacks. We call it footprint, and it aims to represent the involvement of the attacker. Indeed, a structured cyberattack is a sequence of atomic attacks directed towards a goal. By atomic attack we mean vulnerability exploits, such as GPS spoofing, DoS, data injection, etc. (cf III-D) that are steps used by the attacker to reach a goal. Atomic attacks are linked/dependent one to another as they are the necessary steps to reach a goal. This dependence is trivial, as an atomic attack allows the intruder to perform a second attack and so on. It can be an attack providing knowledge about a password, a privilege escalation, a lateral movement, etc. Therefore, we filter attack sequences based on the dependence between two successive events.

In a DES, after the execution of a transition $t \in T$ we have 4 disjoint sets of transitions:

- $S_{00}(t)$: the transitions not enabled before and after the firing of t;
- $S_{01}(t)$: the transitions not enabled before the firing of t and enabled after;
- $S_{10}(t)$: the transitions enabled before the firing of t but not after;
- $S_{11}(t)$: the transitions enabled before the firing of t and still after.

When generating the sequences, we are not interested in $S_{00}(t)$ and $S_{10}(t)$ as transitions within these sets are not enabled. $S_{11}(t)$ and $S_{01}(t)$ are, on the other hand, attacks or system reconfiguration that can be used for the next step. $S_{11}(t)$ are transitions that were enabled before t, thus they are not related to t (they can correspond to another attack path). $S_{01}(t)$ are transitions that were enabled after the firing of t. Thus, transitions from $S_{01}(t)$ depend on t. Therefore, to prioritize the generation of dependent transitions, we introduce the notion of footprint F:

$$F(t_i/t_{i-1}) = \begin{cases} 1 & \text{if } t_i \in S_{11}(t_{i-1}) \\ 0 & \text{if } t_i \in S_{01}(t_{i-1}) \end{cases} \quad (1)$$

TE : Attack on Active Brake Function	TE : Unwanted Braking
UMTS.Use_diagnosis_tool_to_connect_to_the_system_via_internet	UMTS.Use_diagnosis_tool_to_connect_to_the_system_via_internet
UMTS.Download_malicious_code_encrypted_as_an_update	Integrity_Module.Exploit_vulnerability_to_deactivate_integrity_module
Integrity_Module.Integrity_module_fails_to_detect_malicious_code_in_update	UMTS.Download_malicious_code_encrypted_as_an_update
Communication_Unit.Flash_malicious_code_or_malware	Communication_Unit.Forward_malicious_code_encrypted_as_an_update
Communication_Unit.Inject_data_on_backbone_to_exploit_vulnerability	BEM.Flash_Malware_encrypted_as_an_update
Chassis_Safety_Controller.Exploit_CSC_state_handling	BEM.Inject_data_on_backbone_to_exploit_vulnerability
Chassis_Safety_Controller.Corrupt_code_or_incoming_data	Chassis_Safety_Controller.Exploit_CSC_state_handling
Brake_Controller.Automatic_brake_function_degraded	Chassis_Safety_Controller.Process_and_forward_false_neighborhood_brake_notification
	Brake_Controller.Unwanted_activation_of_automatic_brake_function

Figure 3. Example of Sequences Generated for both Top Events

Then, when firing the transitions, we can calculate the footprint of the sequence of length n as the sum of the individual footprints of its transitions.

$$F(\sigma) = F(t_0) + \sum_{i=1}^{n} F(t_i/t_{i-1}),\ \textit{with}\ F(t_0) = 1 \qquad (2)$$

The footprint can be seen as the number of paths explored by the attacker. The initial transition have a footprint $F = 1$ to illustrate that the attacker will explore at least one path. As a result, a sequence with only dependent events will have a footprint $F = 1$ and a sequence with events unrelated executed at a random order will have a large footprint. The next subsection illustrates this definition by filtering the sequences obtained in Section IV-B with a footprint.

B. Illustration of Footprint on the Case Study

An algorithm embedding the notion of footprint shall work almost as any other algorithm embedding cutoffs. The major difference is in the identification of the sets S_{01} and S_{11}. Then, the algorithm will explore a sequence until a safety critical state is reached or the footprint of the sequence reaches the pre-defined boundary.

The footprint has been implemented on SimfiaNeo, which allows to have a clear view on the benefit of the approach. Therefore, we can regenerate the sequences of Section IV-B (length 10) with an additional cutoff on the footprint. The analysis of the unwanted brake activation, with a footprint of value $F \leq 2$, gives 44 minimal critical sequences. If we consider only sequences with footprint $F = 1$, it is lowered to 14 sequences. The attack on the active brake function has 72 sequences, with the criterion $F \leq 2$ and 24 with $F = 1$.

Relation between minimality and footprints: The reduction enabled by the footprint is not always significant: all sequences of footprint $F = 1$ are minimal. In such sequences, all events depend on another and are necessary to achieve the goal. Therefore, when all MCS have a footprint $F = 1$, the generation of minimal sequences is sufficient.

However, having a footprint $F > 1$ implies that independent actions or independent subsequences of dependent actions are part of the scenario. Subsequences are sets of actions fired in a given order, included in a sequence and shorter than the sequence. Then, if a transition independent from the previous is fired, the footprint is incremented. Here, the generator of MCS will suffer from a shuffle between these independent subsequences. This shuffle corresponds to the combinations between events from independent subsequences. For example with three independent subsequences a, b, c, the shuffle between them will be events from every subsection happening in every order, e.g. $a_1b_1c_1a_2$... or $a_1a_2c_1c_2b_1 \ldots$. Thus, a cutoff based on footprint will have a significant effect to remove the shuffle and keep the outputs where the subsequences happen one after another (i.e. $a_1a_2 \ldots a_nb_1 \ldots b_nc_1 \ldots c_n$).

C. Comparison of Cutoffs and Footprints in Sequence Generation

Now that the footprint has been defined and illustrated on the case study, we will compare the result of the analyses performed in Section IV-B and V-B. In this section, we remind how the generation filters differ and analyze the results of the different computations.

Differences between order and footprint: The cutoff on the number of events is used in safety analyses where, with independent failures, the number of events is a good indicator of the likelihood of the sequence. Footprint allows to generate sequences of dependent events. By giving a maximum value, it shows the maximum number of paths that the attack can visit. When considering cyberattacks, filtering on the number of events will miss many likely critical sequences.

If we focus on the exploration, footprints can also be a significant advantage to avoid the shuffle and reduce computational cost. The cutoff will explore all sequences under a given length where footprint will stop the exploration sooner, when too many independent events are fired. There is only one case where the footprint is no better than the number of events. It is when all events are independent, then the two filters are equivalent.

Discussion on the results of the analyses: Table II presents the results obtained when generating the sequences for both unwanted situations and with different parameters. The analyses were performed with the same machine (Intel®Core™ i7-8565U CPU 1.80GHz × 8).

First, we must highlight the major reduction in the number of sequences and computation time, which is achieved via minimality. Then, for both simulations, MCS with footprints

Table II
NUMBER OF SEQUENCES GENERATED AND COMPUTATION TIME WITH DIFFERENT STATE SPACE REDUCTION FOR THE CASE STUDY

Safety-critical event	Output	Length 10	+ Minimality	+ Footprint $F \leq 2$	+ Footprint $F = 1$
Unwanted brake activation	Number of sequences	133429	45	44	14
	Computation time	14min58sec	11min 15sec	1sec	$<$ 1sec
Attack active brake function	Number of sequences	401625	116	72	24
	Computation time	14min35sec	12min 34sec	$<$ 1sec	$<$ 1sec

decreases the number of sequences compared with MCS with cutoffs (45 to 44 and 116 to 72). As we can see in this table, the reduction is not always significant for the reason we evoke in V-B. However, it helps the analysis of the generated sequences by taking off sequences (even with few events) containing random actions. The results still contain shuffle between subsequences of dependent events, but the shuffle is limited by the footprint. For example, with $F = 2$ and with two subsequences a and b, the results will show ab or ba. The most significant reduction obtained with footprint concerns the computation time (from over 10 minutes to a second). This result is very encouraging, as it may improve the scalability of the approach. Indeed, with a lowered computation time, we shall be able to model and generate sequences from larger use cases.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we presented a tooled assessment to evaluate the risks of cyberattacks progression on CPS and their effects on safety. We used the language AltaRica Data-Flow and the graphical tool SimfiaNeo to model CPS and their behavior in the presence of cyberattacks. This tool allows to represent the dynamic behavior of the system and to generate the sequences of actions leading to a safety critical state. Then, to overcome the state-space explosion in the sequence generation, we propose a cutoff called footprint. This cutoff benefits from the dependent nature of cybersecurity events to lower the state space and save computation time. Finally, we illustrate the modeling and the footprint with an automotive case study, and discuss the results of the analysis.

This approach is a significant improvement towards the automation of the security risk analysis with the use of a formal language. It shows how atomic attacks identified by engineers can be used altogether in a complex cyberattack. Therefore, this work will allow to consider larger systems without the fear of missing potentially catastrophic scenarios.

The continuity of this work would be to consider the likelihood of each event and use them to assess the likelihood of the sequences. This would allow complying with industrial regulations such as ISO 27001 [39]. Finally, we shall evaluate the scalability of the model to check if we can assess larger use cases or assess them at a lower abstraction level.

REFERENCES

[1] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, "Modelling with Generalized Stochastic Petri Nets," *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, no. 2, p. 2, Aug. 1998. [Online]. Available: https://doi.org/10.1145/288197.581193

[2] M. Bouissou and Y. Lefebvre, "A path-based algorithm to evaluate asymptotic unavailability for large Markov models," in *Proceedings of the Reliability and Maintainability Symposium*, Feb. 2002, pp. 32–39. [Online]. Available: https://doi.org/10.1109/RAMS.2002.981616

[3] M. Batteux, T. Prosvirnova, and A. Rauzy, "Altarica 3.0 in 10 modeling patterns," *International Journal of Critical Computer-Based Systems*, vol. 9, no. 1–2, pp. 133–165, 2019. [Online]. Available: https://doi.org/10.1504/IJCCBS.2019.098809

[4] J.-M. Thiriet and S. Mocanu, "Some Considerations on Dependability Issues and Cyber-Security of Cyber-Physical Systems," in *The 7 th IEEE International Conference on Smart Communications in Network Technologies (SACONET'18)*, Oct. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01909025

[5] H. Bagheri, E. Kang, S. Malek, and D. Jackson, "A formal approach for detection of security flaws in the android permission system," *Formal Aspects of Computing*, vol. 30, no. 5, pp. 525–544, Sep. 2018. [Online]. Available: https://doi.org/10.1007/s00165-017-0445-z

[6] L. Sassaman, M. L. Patterson, S. Bratus, and M. E. Locasto, "Security Applications of Formal Language Theory," *IEEE Systems Journal*, vol. 7, no. 3, pp. 489–500, Sep. 2013.

[7] M. Avalle, A. Pironti, and R. Sisto, "Formal verification of security protocol implementations: a survey," *Formal Aspects of Computing*, vol. 26, no. 1, pp. 99–123, Jan. 2014. [Online]. Available: https://dl.acm.org/doi/10.1007/s00165-012-0269-9

[8] M. Roggenbach, A. Cerone, B.-H. Schlingloff, G. Schneider, and S. A. Shaikh, "Formal Verification of Security Protocols," in *Formal Methods for Software Engineering: Languages, Methods, Application Domains*, ser. Texts in Theoretical Computer Science. An EATCS Series. Cham: Springer International Publishing, 2022, pp. 395–451. [Online]. Available: https://doi.org/10.1007/978-3-030-38800-3_8

[9] D. J. Bodeau, C. D. McCollum, and D. B. Fox, "Cyber Threat Modeling: Survey, Assessment, and Representative Framework," MITRE CORP MCLEAN VA MCLEAN, Tech. Rep., Apr. 2018. [Online]. Available: https://apps.dtic.mil/sti/citations/AD1108051

[10] R. Lanotte, M. Merro, R. Muradore, and L. Viganò, "A Formal Approach to Cyber-Physical Attacks," *arXiv:1611.01377 [cs]*, Apr. 2017. [Online]. Available: http://arxiv.org/abs/1611.01377

[11] M. Burmester, E. Magkos, and V. Chrissikopoulos, "Modeling security in cyber–physical systems," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3-4, pp. 118–126, Dec. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1874548212000443

[12] P. Nguyen, S. Wang, and T. Yue, "Model-Based Security Engineering for Cyber-Physical Systems: A Systematic Mapping Study," *Information and Software Technology*, vol. 83, Nov. 2016.

[13] P. H. Nguyen, M. Kramer, J. Klein, and Y. L. Traon, "An extensive systematic review on the Model-Driven Development of secure systems," *Information and Software Technology*, vol. 68, pp. 62–81, Dec. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584915001482

[14] R. Kumar, "Truth or Dare: Quantitative security risk analysis using attack trees," Ph.D. dissertation, University of Twente, Oct. 2018. [Online]. Available: https://doi.org/10.3990/1.9789036546256

[15] S. Mauw and M. Oostdijk, "Foundations of Attack Trees," in *Information Security and Cryptology - ICISC 2005*, Berlin, Heidelberg, 2006, vol. 3935, pp. 186–198. [Online]. Available: http://link.springer.com/10.1007/11734727_17

[16] W. Widel, M. Audinot, B. Fila, and S. Pinchinat, "Beyond 2014: Formal Methods for Attack Tree–based Security Modeling," *ACM Computing*

Surveys, vol. 52, no. 4, pp. 75:1–75:36, Aug. 2019. [Online]. Available: https://doi.org/10.1145/3331524

[17] A. Tantawy, S. Abdelwahed, A. Erradi, and K. Shaban, "Model-based risk assessment for cyber physical systems security," *Computers & Security*, vol. 96, p. 101864, Sep. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740482030136X

[18] A. Roy, D. S. Kim, and K. Trivedi, "Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees," *Security and Communication Networks*, vol. 5, Aug. 2012. [Online]. Available: https://doi.org/10.1002/sec.299

[19] E. André, D. Lime, M. Ramparison, and M. Stoelinga, "Parametric Analyses of Attack-fault Trees," *Fundamenta Informaticae*, vol. 182, no. 1, pp. 69 – 94, Sep. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03483440

[20] S. Choi, J.-H. Yun, and B.-G. Min, "Probabilistic Attack Sequence Generation and Execution Based on MITRE ATT&CK for ICS Datasets," in *Cyber Security Experimentation and Test Workshop*, ser. CSET '21, New York, NY, USA, Aug. 2021, pp. 41–48. [Online]. Available: https://doi.org/10.1145/3474718.3474722

[21] R. Kumar, D. Guck, and M. Stoelinga, "Time Dependent Analysis with Dynamic Counter Measure Trees," Sep. 2015. [Online]. Available: http://arxiv.org/abs/1510.00050

[22] N. Idika and B. Bhargava, "Extending Attack Graph-Based Security Metrics and Aggregating Their Application," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 75–85, Jan. 2012. [Online]. Available: https://ieeexplore.ieee.org/document/5611550

[23] N. Ghosh and S. K. Ghosh, "A planner-based approach to generate and analyze minimal attack graph," *Applied Intelligence*, vol. 36, no. 2, pp. 369–390, Mar. 2012. [Online]. Available: http://link.springer.com/10.1007/s10489-010-0266-8

[24] Y. Zhang, L. Wang, Y. Xiang, and C.-W. Ten, "Power System Reliability Evaluation With SCADA Cybersecurity Considerations," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1707–1721, Jul. 2015. [Online]. Available: https://doi.org/10.1109/TSG.2015.2396994

[25] E. Bourget, "Diagnosing accidental and malicious events in industrial control systems," phdthesis, Ecole nationale supérieure Mines-Télécom Atlantique, Jun. 2020. [Online]. Available: https://tel.archives-ouvertes.fr/tel-03169404

[26] D. Miller, R. Alford, A. Applebaum, H. Foster, C. Little, and B. E. Strom, "Automated Adversary Emulation: A Case for Planning and Acting with Unknowns," *MITRE*, Jun. 2018. [Online]. Available: https://www.mitre.org/publications/technical-papers/automated-adversary-emulation-a-case-for-planning-and-acting-with

[27] J. D. Yoo, E. Park, G. Lee, M. K. Ahn, D. Kim, S. Seo, and H. K. Kim, "Cyber Attack and Defense Emulation Agents," *Applied Sciences*, vol. 10, no. 6, p. 2140, Jan. 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/6/2140

[28] S. Ullah, S. Shetty, A. Nayak, A. Hassanzadeh, and K. Hasan, "Cyber Threat Analysis Based on Characterizing Adversarial Behavior for Energy Delivery System," in *Security and Privacy in Communication Networks*. Cham: Springer International Publishing, 2019, vol. 305, pp. 146–160. [Online]. Available: http://link.springer.com/10.1007/978-3-030-37231-6_8

[29] M. Drasar, S. Moskal, S. Yang, and P. Zat'ko, "Session-level Adversary Intent-Driven Cyberattack Simulator," in *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. Prague, Czech Republic: IEEE, Sep. 2020, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/9213690/

[30] C. J. Deloglos, C. R. Elks, and A. Tantawy, "An attacker modeling framework for the assessment of cyber-physical systems security," *CoRR*, 2020. [Online]. Available: https://arxiv.org/abs/2006.03930

[31] S. Kriaa, "Joint safety and security modeling for risk assessment in cyber physical systems," Ph.D. dissertation, Centrale Supelec, 2016. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01318118/document

[32] A. Wasicek, P. Derler, and E. A. Lee, "Aspect-Oriented Modeling of Attacks in Automotive Cyber-Physical Systems," in *Proceedings of the 51st Annual Design Automation Conference*. San Francisco, CA, USA: ACM Press, 2014, pp. 1–6. [Online]. Available: https://doi.org/10.1145/2593069.2593095

[33] E. Kang, S. Adepu, D. Jackson, and A. P. Mathur, "Model-Based Security Analysis of a Water Treatment System," in *Proceedings of the 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems*. Austin, TX, USA: ACM Press, 2016, pp. 22–28. [Online]. Available: https://doi.org/10.1145/2897035.2897041

[34] EVITA Project, "Evita: E-safety vehicle intrusion protected applications," 2011. [Online]. Available: https://www.evita-project.org/

[35] E. Kelling, M. Friedewald, T. Leimbach, M. Menzel, P. Säger, H. Seudié, and B. Weyl, "Specification and evaluation of e-security relevant use cases. Deliverable D2.1: EVITA. E-Safety Vehicle Intrusion Protected Applications," *Fraunhofer ISI*, May 2022. [Online]. Available: https://www.researchgate.net/publication/46307752_Security_requirements_for_automotive_on-board_networks_based_on_dark-side_scenarios_Deliverable_D23_EVITA_E-safety_vehicle_intrusion_protected_applications

[36] A. Ruddle, D. Ward, B. Weyl, M. S. IDREES, Y. Roudier, M. Friedewald, T. Leimbach, A. Fuchs, S. Gurgens, O. Henniger, R. Roland, M. Ritscher, H. Broberg, L. Apvrille, R. Pacalet, and G. Pedroza, "Security requirements for automotive on-board networks based on dark-side scenarios, Deliverable D2.3," Telecom ParisTech, Contract EVITA, Mar. 2010. [Online]. Available: https://hal.telecom-paris.fr/hal-02286288

[37] M. Boiteau, Y. Dutuit, and A. Rauzy, "The AltaRica Data-Flow Language in Use: Modeling of Production Availability of a MultiStates System," *Reliability Engineering & System Safety*, vol. 91, pp. 747–755, Jul 2006. [Online]. Available: https://doi.org/10.1016/j.ress.2004.12.004

[38] A. Rauzy, "Mode automata and their compilation into fault trees," *Reliability Engineering & System Safety*, vol. 78, no. 1, pp. 1–12, Oct. 2002. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S095183200200042X

[39] "ISO/IEC 27001:2013." [Online]. Available: https://www.iso.org/cms/render/live/fr/sites/isoorg/contents/data/standard/05/45/54534.html