CHALMERS
UNIVERSITY OF TECHNOLOGY

Proposing HEAVENS 2.0 – an automotive risk assessment model

Proposing HEAVENS 2.0 – an automotive risk assessment model

Aljoscha Lautenbach
aljoscha@chalmers.se
Chalmers University of Technology
Gothenburg, Sweden
Aljoscha.Lautenbach@evidente.se
Evidente AB
Gothenburg, Sweden

Magnus Almgren
Tomas Olovsson
magnus.almgren@chalmers.se
tomas.olovsson@chalmers.se
Chalmers University of Technology
Gothenburg, Sweden

ABSTRACT

Risk-based security models have seen a steady rise in popularity over the last decades, and several security risk assessment models have been proposed for the automotive industry. The new UN vehicle regulation 155 on cybersecurity provisions for vehicle type approval, as part of the 1958 agreement on vehicle harmonization, mandates the use of risk assessment to mitigate cybersecurity risks and is expected to be adopted into national laws in 54 countries within 1 to 3 years. This new legislation will also apply to autonomous vehicles. The automotive cybersecurity engineering standard ISO/SAE 21434 is seen as a way to fulfill the new UN legislation, so we can expect quick and wide industry adoption. One risk assessment model that has gained some popularity and is in active use in several companies is the HEAVENS model, but since ISO/SAE 21434 introduces additional requirements on the risk assessment process, the original HEAVENS model does not fulfill the standard.

In this paper, we investigate the gap between the HEAVENS risk assessment model and ISO/SAE 21434, and we identify and propose 12 model updates to HEAVENS to close this gap. We also discuss identified weaknesses of the HEAVENS risk assessment model and propose 5 additional model updates to overcome them. In accordance with these 17 identified model updates, we propose HEAVENS 2.0, a new risk assessment model based on HEAVENS which is fully compliant with ISO/SAE 21434.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; *Dependable and fault-tolerant systems and networks*; • **Security and privacy** → Security requirements; **Systems security**; **Embedded systems security**.

KEYWORDS

ISO/SAE 21434, UNECE regulation 155, Automotive, Risk Assessment, Threat Analysis, TARA

CSCS '21, November 30, 2021, Ingolstadt, Germany

ACM Reference Format:
Aljoscha Lautenbach, Magnus Almgren, and Tomas Olovsson. 2021. Proposing HEAVENS 2.0 – an automotive risk assessment model. In *Computer Science in Cars Symposium (CSCS '21), November 30, 2021, Ingolstadt, Germany.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3488904.3493378

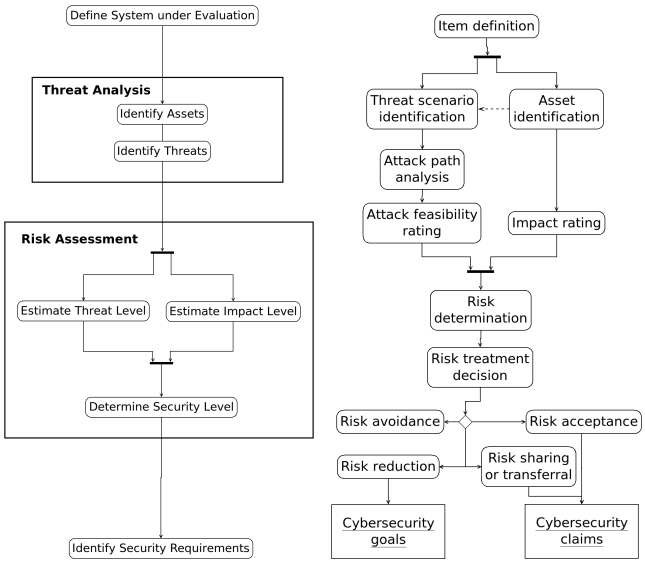
1 INTRODUCTION

Modern vehicles are typical cyber-physical systems: they have a large number of interconnected electrical and electronic systems with complex sensor and actuator interactions. In addition to having complex internal interfaces, vehicles also have an increasing number of external interfaces, and autonomous driving is accelerating this trend even further. With this rise of connectivity and complexity, the need for security has been rising in lock-step, a fact that has been proven consistently for over 15 years [1, 7, 14, 15, 35].

As this need for security is slowly being acknowledged in automotive and legislative circles, new standards and regulations have started to emerge. In 2016, SAE published the "SAE J3061 Cybersecurity Guidebook for Cyber-Physical Vehicle Systems" [19], which was an early effort to document industry best-practices around automotive cybersecurity and to set a common level of expectation. Shortly after, a joint task force of SAE and ISO was formed to work on the new standard "ISO/SAE 21434 Road Vehicles - Cybersecurity Engineering" to supersede J3061, which was officially published in August 2021 [5].

Both SAE J3061 and ISO/SAE 21434 cover various aspects of the entire life-cycle of a vehicle, but significant emphasis is put on threat analysis and risk assessment (TARA) which is supposed to be performed at least once in the concept phase. The goal of threat analysis and risk assessment is to identify and rate potential threats in order to determine which threats need to be mitigated and what level of mitigation is required. Several models and frameworks have been proposed for automotive TARA, one of which has become known as the HEAVENS security model [2, 3]. HEAVENS is in use in several organizations and was explicitly referenced in SAE J3061 as a possible approach to automotive TARA, but the risk assessment framework that is outlined in ISO/SAE 21434 is rather generic and has a slightly different workflow. Moreover, as the industrial experience with TARA has grown, several practical problems have been found with the HEAVENS security model. Nevertheless, with minor modifications HEAVENS can fulfill the requirements of the standard, and adds value by providing a clear methodology to follow.

In this paper, we therefore propose HEAVENS 2.0 which is compatible with the upcoming ISO/SAE standard 21434, and which addresses the problems that have been found with the original



(a) HEAVENS 1.0 workflow (b) ISO/SAE 21434 TARA workflow

Figure 1: Juxtaposition of HEAVENS 1.0 and ISO/SAE 21434 workflows

HEAVENS security model (we will refer to it as HEAVENS 1.0 from now on). To this end, we make the following contributions:

- We investigate the gap between HEAVENS 1.0 and the risk assessment model mandated by ISO/SAE 21434 and enumerate 12 model updates to HEAVENS 1.0 which are required to close this gap
- We discuss identified weaknesses of HEAVENS 1.0 and propose 5 additional model updates to overcome them
- We present HEAVENS 2.0 which is a new version that incorporates the previously identified model updates
- We demonstrate the viability of HEAVENS 2.0 with an example.

Note that although the risk assessment methodologies of HEAVENS 1.0 and 2.0 were specifically designed for the automotive industry, with minor calibrations of the parameters they can also be applied to other industries with similar characteristics, such as medical devices or industrial systems.

2 DEFINING THE GAP

Among other requirements, ISO/SAE 21434 mandates a detailed risk assessment process for automotive development projects without prescribing a specific methodology. In this section, we perform a gap analysis by comparing the HEAVENS 1.0 workflow and terminology with the risk assessment requirements in ISO/SAE 21434, and we discuss necessary model adjustments based on this analysis. We enumerate these model updates, and prefix them either with "G" for an identified gap to ISO/SAE 21434, or with "P" for an identified problem of HEAVENS 1.0 (see section 3). We discuss how to address the identified updates in section 4.

A brief summary of HEAVENS 1.0 is provided in appendix B and its workflow is depicted in figure 1a, but for the full description please refer to the original paper [3]. The ISO/SAE 21434 risk assessment workflow is depicted in figure 1b.

It is obvious in figure 1 that the terminologies do not match. This leads us to the first required update of HEAVENS 1.0: **`UPDATE G1 – Align terminology`**. Also evident in figure 1 is that ISO/SAE 21434 makes no distinction of the threat analysis and risk assessment phases. The lines between threat analysis and risk assessment are blurry and one could argue that they are indistinguishable in this context. It therefore seems prudent to combine the two phases: **`UPDATE G2 – Merge threat analysis and risk assessment phases`**.

In the following, we compare the workflows and corresponding activities, and we highlight when the HEAVENS 1.0 workflow or activities do not fulfill the requirements of ISO/SAE 21434. Both workflows begin with a system or item definition, and continue with asset identification. However, asset identification in ISO/SAE 21434 includes a new sub-activity, damage scenario identification, that did not exist in HEAVENS 1.0: **`UPDATE G3 – Include damage scenario identification`**.

In figure 1b, the path to "attack feasibility rating" includes the creation of threat scenarios and attack paths. As in HEAVENS 1.0, STRIDE can be used for threat scenario identification. However, attack path analysis is a new activity that must be included: **`UPDATE G4 – Include attack path analysis`**. It is the attack paths which are rated for attack feasibility, rather than rating the threat likelihood per asset as HEAVENS 1.0 does. This has the advantage of being more realistic since the entire attack chain has to be considered, but the downside is that it takes extra effort to create attack paths for every threat scenario. While the standard gives recommendations for the attack feasibility rating methodologies to use, including the attack potential based approach that is used in HEAVENS 1.0 [13], no specific methodology is required. The standard does however require a specific number and specific names for the levels of the attack feasibility rating: **`UPDATE G5 – Adjust threat levels`**.

For the impact rating, an important difference that is not reflected in the workflow is a shift in stakeholder perspective: HEAVENS 1.0 focuses primarily on the OEM perspective, whereas ISO/SAE 21434 specifies the road user as the primary stakeholder. This shift has a particularly strong effect on the impact rating, but it also informs which damage and threat scenarios to focus on. However, ISO/SAE 21434 clearly specifies that additional stakeholders can be defined. The rationale for making the road user the primary stakeholder is that the result of the TARA should be similar among all performing organizations, and having the end-user in mind facilitates that goal: **`UPDATE G6 – Shift the stakeholder perspective`**.

ISO/SAE 21434 does not mandate a specific methodology for the impact rating, but it includes some specific requirements such as which impact categories to consider, namely safety, financial, operational and privacy. These categories are aligned with HEAVENS 1.0 with one notable difference: In HEAVENS 1.0, the last impact category combined privacy and legislative impact, so an adjustment is needed: **`UPDATE G7 – Remove legislative impact parameter`**. In addition to this minor update, an adjustment of the impact levels

is also required, specifically changing the level names and reducing the number of levels: **`UPDATE G8 – Adjust impact levels`**.

Finally, HEAVENS 1.0 does not clearly separate the risk from the resulting security level. In other words, risk is not calculated separately, the outcome of the model is the security level. In contrast, ISO/SAE 21434 has no concept of a security level but uses a risk value to describe risk: **`UPDATE G9 – Rename security level to risk value`**. Similar to the threat and impact levels, the allowable risk levels are also fixed in ISO/SAE 21434, and thus require a change: **`UPDATE G10 – Adjust security levels`**. The last step in HEAVENS 1.0 was to identify security requirements, but ISO/SAE 21434 has a slightly different, albeit similar, approach. Once the risk for a particular threat scenario is known, the risk treatment decision determines which action to take: **`UPDATE G11 – Include risk treatment decision and resulting actions`**. If a risk is accepted or shared/transferred, a cybersecurity claim must be written and if risk reduction is necessary, cybersecurity goals which define high-level requirements must be created: **`UPDATE G12 – Include cybersecurity claims and goals`**. This covers all the model updates required to align HEAVENS 1.0 with ISO/SAE 21434.

3 PROBLEMS OF HEAVENS 1.0

In addition to the gap with ISO/SAE 21434, there are specific problems with HEAVENS 1.0 that should be addressed. Implementations in industrial projects have shown that certain aspects are not very practical or have lead to difficulties. The identified problems are based on qualitative feedback from industry practitioners who have used HEAVENS 1.0 in real-world projects. In this section we highlight and categorize these problems, and we propose model adjustments and recommendations.

Sandberg, Bokesand and Thorsson have identified several issues with practical applications of HEAVENS 1.0 [20]. We summarize their findings and add additional observations, and we mark our observations as [NEW] to clearly distinguish the two.

The encountered problems can be grouped and summarized as follows:

(1) **Learnability**
  (a) Counter-intuitive threat values [NEW]
(2) **Model customization**
  (a) Lack of parameter normalization [NEW]
(3) **Process efficiency and accuracy**
  (a) Wasted effort [NEW]
  (b) Unclear parameter guidance [20]

Since these problems concern details of the methodology, any changes to address them have no effect on the model's applicability to ISO/SAE 21434.

3.1 Learnability

A declared goal for HEAVENS 1.0 was to be easy to understand and to apply. Experience has shown that this holds mostly true, but there is one frequent source of confusion: the inverse relationship of threat value and threat level, which is perceived as counter-intuitive.

Counter-intuitive threat values. That higher threat levels have a lower threat value is a common source of confusion when introducing HEAVENS 1.0 to TARA participants. The original reasoning for this choice was that the worst-case values can not be made worse, but there might be room for additional levels in the opposite direction. However, it seems unlikely that additional levels are needed, so aligning the model with people's intuition should take precedence. This is a minor process hurdle, but it is worth to flatten the learning curve. The solution is to reverse the parameter levels and to adjust the threat level sum accordingly: **`UPDATE P1 – Inverse threat level sum`**.

3.2 Model customization

Another aspect of HEAVENS 1.0 is that it should be customizable to particular project needs. For instance, one might want to re-add the "elapsed time" parameter for the attack potential calculation, or to add a new impact category, such as financial impact for the OEM. Unfortunately, one aspect in particular, namely the lack of normalization, makes this hard.

Lack of parameter normalization. Since the threat level parameter and the impact level parameter sums are not normalized, introducing new parameters requires manual adjustments to the parameter sum tables, which is time consuming (cf. tables 14 and 16 in appendix B). The solution is to normalize the parameter sums: **`UPDATE P2 – Normalize threat level parameter sum`** and **`UPDATE P3 – Normalize impact level parameter sum`**.

3.3 Process efficiency and accuracy

According to Sandberg, Bokesand and Thorsson [20], HEAVENS 1.0 has two main problems: (1) speed in performing the TARA, and (2) consistency in the results of the TARA. In other words, process efficiency and accuracy are problematic, due to wasted effort and unclear parameter guidance.

Wasted effort. One of the main difficulties in strictly applying HEAVENS 1.0 is the sheer volume of asset/threat pairs that need to be evaluated. Without a pre-filtering mechanism, every threat/asset pair needs to go through the entire TARA before it can be discarded due to a low risk. This can require a lot of effort for threats which have a low impact, thus leading to wasted effort.

A possible solution would be to pre-screen the threats after the impact level estimation to weed out low impact threats. However, completely ignoring the likelihood of the threat might give too much weight to threats of high impact which are unlikely to be realized. To attenuate this problem, we propose to use an approximation for the threat level in a first iteration, namely to use only accessibility as a gauge for the threat level. In other words, if an attack requires physical access, the threat level is low, and if an attack can be performed remotely the threat level is high. This saves time in a first round of estimation and allows to look at the high impact threats first. **`UPDATE P4 – Simplify threat level estimation for threat pre-screening`**.

Unclear parameter guidance. Since a typical TARA is performed by a group of domain and security experts, consensus must be reached in order to complete the TARA. But as Sandberg et al. point out, unclear guidance on parameter values can lead to long

discussions which in turn lead to delays and inconsistencies due to varying interpretations by different groups. Clear and unambiguous guidance for parameter values is therefore critical for a smooth and consistent process. Sandberg et al. identify the parameters "window of opportunity" and "financial impact" as particular problematic.

Since "window of opportunity" has two main dimensions, they propose to further split this parameter into two sub-parameters, specifically "access means" and "exposure time" [20]. "Access means" represents the physical access dimension, meaning how close an attacker needs to be in order to perform an attack, whereas "exposure time" represents the time dimension, i.e., how long an attacker has access to the asset to perform an attack. This requires an extra step to map the two sub-parameters into a single "window of opportunity" value, but it clarifies the interpretation and therefore minimizes the need for discussion. We propose to follow this recommendation. **`UPDATE P5 – Add sub-parameters for "window of opportunity"`**.

The difficulty with the financial impact parameter is that many factors play into financial impact, such as the other impact parameters of safety, privacy and operational, as well as other direct and indirect factors such as expected repair costs, loss of intellectual property, loss of reputation, etc. Therefore, Sandberg et al. propose to define sub-parameters as needed, with clear guidance on how to combine them into a consistent view. However, since the deconstruction of financial impact can vary widely between organizations they do not propose any particular sub-parameters in general, and for the same reason, we will not address this either.

4 HEAVENS 2.0

In this section we present the new model HEAVENS 2.0 which incorporates the model updates identified in sections 2 and 3. The updates are summarized in Table 11 in appendix A.

The proposed model updates lead to the new workflow of HEAVENS 2.0, which is depicted in figure 2. The figure highlights the changes from HEAVENS 1.0 through color coding, i.e., whether a particular activity is unchanged, has been modified or is completely new. Only three activities remain unchanged, all other activities contain at least one update or are new. For easier orientation, figure 2 also includes a section reference map that indicates which activities are discussed in which sub-section. We first discuss the model updates which involve the workflow as a whole, and then discuss the workflow activities one by one in the following subsections, including any relevant model updates.

`UPDATE G1`, updating the terminology, is clearly a cross-activity concern. Since we expect ISO/SAE 21434 to see widespread adoption in the industry, we chose to adopt its terminology as closely as possible. Table 1 shows the terminology mapping.

`UPDATE G2`, merging the threat analysis and risk assessment phases, is already included in figure 2, and the update's rationale has been given in section 2. The remaining updates concern specific activities, so we continue with a detailed walk-through of the workflow and discuss the updates when appropriate.

4.1 Item definition

The only, yet vital, input to the HEAVENS 2.0 TARA is the item definition, an output of the item definition activity. Except for the

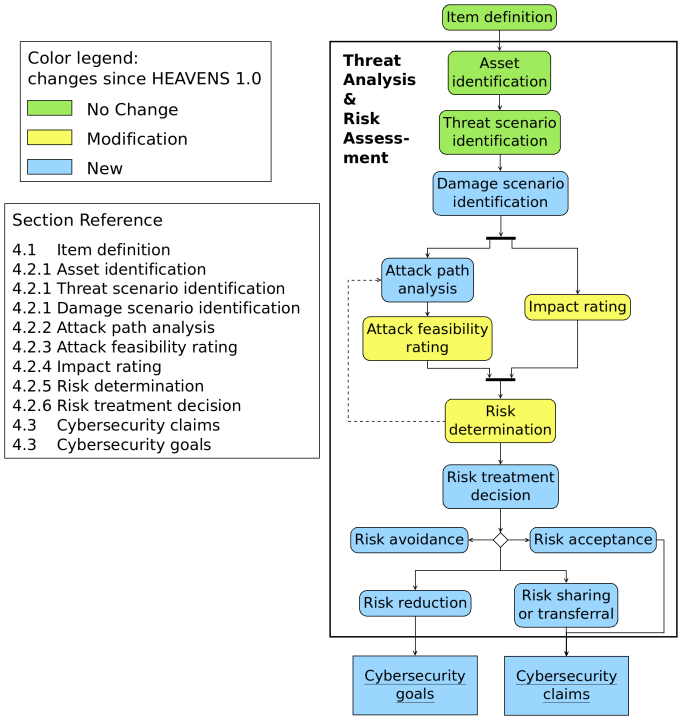


Figure 2: HEAVENS 2.0 workflow

Table 1: Terminology mapping from HEAVENS 1.0 to HEAVENS 2.0

HEAVENS 1.0	HEAVENS 2.0 (& ISO/SAE 21434)
Asset/threat pair	Attack
High-level security requirement	Cybersecurity goal
Impact level	Impact rating
Security level	Risk value
System under evaluation	Item
Threat	Threat scenario
Threat level	Attack feasibility rating

name, the activity remains unchanged from HEAVENS 1.0. In general, the item definition includes all parts that are required to start work on the item, such as the definition of the item boundary, the item function, the preliminary architecture as well as the operational environment and relevant assumptions about the item and its environment.

Note that in order to use STRIDE for asset identification and threat scenario identification later, at least an initial data flow diagram (DFD), which includes the most important entities and data flows of the item, must be included in the item definition.

A vital aspect of the item definition is choosing the correct level of abstraction because it is very easy to get lost in unnecessary details. In a TARA the goal is to identify high-level high-priority threats, we are only interested in the overall data flow, not in every single CAN signal or message. Pinpointing the exact CAN signal

that needs to be protected can be left to the technical requirements stage. Since most vehicles include tens of thousands of different CAN signals the level of abstraction is crucial to get right.

4.2 Threat analysis and risk assessment

Figure 2 shows that assets, threat scenarios, damage scenarios, and attack paths are identified in the initial phases. Once that has been accomplished, the attack paths are rated for attack feasibility, and the damage scenarios are rated for impact. When the attack feasibility rating and the impact rating have been estimated, the overall risk for the threat scenarios can be determined, after which a treatment decision has to be made. Each of these steps is covered in detail in the following subsections.

4.2.1 Asset, threat scenario and damage scenario identification. Asset identification works the same as in HEAVENS 1.0, namely by either manual inspection of the item definition or by automated identification of assets in a data flow diagram (DFD). Threat scenarios describe a set of actions that lead to one or more damage scenarios, where damage scenarios specify the adverse consequences of an attack; in other words they specify the result of an attack. This distinction introduced in ISO/SAE 21434 facilitates a clear separation of actions and consequences.

As can be seen in figure 2, threat scenario identification does not need to be updated: STRIDE is used to enumerate threats on the assets based on a DFD, which is systematic and can be automated. In accordance with `UPDATE G3`, damage scenarios have been added to HEAVENS 2.0 (figure 2). Note that rather than including the identification of damage scenarios in asset identification, we chose to make it a separate activity for the following reasons: (1) we believe that these activities are sufficiently different to be split into two different activities, (2) this is a more explicit change which might help people familiar with HEAVENS 1.0, and most important (3) by enumerating all threat scenarios first, we can take a systematic approach to identifying damage scenarios, namely through identifying and grouping the consequences of each threat scenario. This leads to a high coverage of threat and damage scenarios and makes it less likely that important scenarios are missed.

4.2.2 Attack path analysis. As per `UPDATE G4`, an activity for attack path analysis has been added. Its aim is to identify the possible attack paths which might realize the threat scenario in question. The advantage is a more accurate attack feasibility rating, because the entire attack chain has to be considered. The downside is that the TARA becomes significantly more time consuming, but since ISO/SAE 21434 requires an attack path analysis, this can not be avoided. ISO/SAE 21434 outlines several alternative methodologies for attack path analysis, but for HEAVENS 2.0 we advocate the use of attack trees, since they are both versatile and well-established. Attack trees also have the advantage that they are quite modular and sub-trees for sub-goals can potentially be reused between components, and help to build a more holistic threat model. Finally, they are similar to fault-trees which are common in safety modeling, which further facilitates communication between safety and security engineers.

Since damage scenarios and threat scenarios have already been identified at this stage, the first step of the attack tree construction is trivial: every damage scenario has a corresponding attack tree in which the damage scenario forms the root of the tree. The threat scenarios which lead to those damage scenarios are the direct children of the root node. Depending on the specificity of the threat scenario, it can be split into several nodes, or new nodes can be constructed on a path to an external facing interface which may be the entry point for an attack. Every attack tree can be constructed this way. The total set of attack paths is then the set of unique paths from leaf node to root node for all attack trees.

4.2.3 Attack feasibility rating. When the set of attack paths has been identified, each attack path should receive an attack feasibility rating. Only one model update needs to be applied to close the gap to ISO/SAE 21434, but four additional updates will be applied to address some of the problems of HEAVENS 1.0. The attack-potential based approach we advocate is based on the attack potential calculation that is part of vulnerability assessment presented in appendix B.4 of the common methodology for information technology security evaluation [13], which is also standardized as ISO/IEC 18045 [4]. This is the same attack potential calculation that is used in HEAVENS 1.0 to determine the threat level, with two exceptions: (1) the parameter values in HEAVENS 1.0 are weighted and use a different scale, and (2) the parameter "elapsed time" is excluded in HEAVENS 1.0 because it can be argued that it is implicitly expressed in the combination of the other parameters, "Expertise", "Knowledge of the [item]", "Window of opportunity" and "Equipment". A summary explanation of the parameters can be found in appendix B.

We begin by addressing `UPDATE P5`, adding sub-parameters to "window of opportunity" to enable a more consistent rating. We follow the recommendations of Sandberg et al. [20] with minor modifications. They propose to use "access means" and "asset exposure time" as sub-parameters, where access means refers to what kind of access is required, physical or remote in different variations, and "asset exposure time" is about the amount of time that a potential attacker has to perform the attack. Tables 2 and 3 show the sub-parameter levels with explanations, and table 4 illustrates how to combine them. While superficially this might look like additional work, it will help to minimize discussions during the TARA, thus actually saving time and effort, and it will make the results more consistent.

Let us address `UPDATE P2` next, i.e., normalizing the attack feasibility calculation. In HEAVENS 1.0 the formula for the attack feasibility calculation is

$$A_{sum} = w_x a_x + w_k a_k + w_w a_w + w_e a_e \tag{1}$$

where w_i and a_i are the weight and the estimated attack feasibility rating of parameter i, and the indices x, k, w, e stand for the four parameters, expertise, knowledge of item, window of opportunity and equipment, respectively. Clearly, adding another parameter will increase the total sum of the calculation, so that the table for assigning the correct attack feasibility rating no longer applies (cf. table 14 in appendix B). The parameter values range from 0 to 3 (cf. table 13 in appendix B), so that the sum can be normalized as follows:

$$A_{nsum} = \frac{w_x a_x + w_k a_k + w_w a_w + w_e a_e}{3 * (w_x + w_k + w_w + w_e)} \tag{2}$$

Table 2: Levels for sub-parameter "access means" in ascending criticality [20]

Level	Explanation	Examples
Physical 1 - component disassembly	Some disassembly of a vehicle component with electronic tools is needed	Any type of low level physical access to read or control a components state, such as attaching a hardware debugger to an electronic control unit (ECU), using a flash reader, etc.
Physical 2 - component access	Some disassembly of the vehicle body with physical tools is needed	Installation or replacement of components, or attaching to a network bus that is otherwise unreachable
Physical 3 - no disassembly	Physical access to the vehicle interior or exterior is needed	Connecting to the OBD-II port, NFC, USB, etc.
Remote 1 - vehicle proximity	Access to a local vehicle network is needed	Bluetooth, Wi-Fi, wireless sensors, V2X, etc.
Remote 2 - anywhere	Remote Internet or telecommunication access is needed	Remote access through the telecommunication network or an external access point

Table 3: Levels for sub-parameter "asset exposure time" [20]

Level	Explanation	Examples
Rare	A single rare moment of exposure that cannot be triggered by the attacker	Factory programming of a specific component, installation of a new component in a workshop, pairing of immobilizer and key fob, etc.
Sporadic	A sporadic moment of exposure that cannot be triggered by the attacker	Certain start-up events, sporadic incoming remote connections, diagnostic tests, infrequent state transitions, etc.
Frequent	A frequent moment of exposure that cannot be triggered by the attacker	Vehicle functions that are often active, such as specific infotainment applications, normal operational states for ECUs, etc.
Unlimited	An unlimited moment of exposure, or one that can be triggered by the attacker	Vehicle functions that are always active or can be activated by an attacker, such as sensors, Bluetooth receivers, wireless gateways, diagnostics servers, etc.

Table 4: Deriving the "window of opportunity" level from the sub-parameters [20]

	Physical 1 (comp. disassembly)	**Physical 2** (comp. access)	**Physical 3** (no disassembly)	**Remote 1** (proximity)	**Remote 2** (anywhere)
Rare	Small	Small	Small	Small	Medium
Sporadic	Small	Small	Small	Medium	Large
Frequent	Small	Small	Medium	Large	Unlimited
Unlimited	Small	Medium	Large	Large	Unlimited

or more generally

$$A_{nsum} = \frac{\sum_i^n w_i a_i}{3 * \sum_j^n (w_j)} \tag{3}$$

where n is the number of parameters. If all the parameters have equal weight, this simplifies to:

$$A_{nsum} = \frac{\sum_i^n a_i}{3 * n} \tag{4}$$

The resulting sum will always be between 0 and 1, so that we can define a new table that still applies even after additional model parameters are added.

Before presenting the new attack feasibility calculation table, let us address the other two updates which affect that table, namely `UPDATE P1`, to reverse the parameter values, and `UPDATE G5`, to adjust the number of attack feasibility levels. Reversing the parameter values is trivial; the result is shown in table 5. Since `UPDATE G5` requires lowering the number of levels from five to four, we opt to present a new table entirely rather than first normalizing the old table (cf. table 14 in appendix B) and then reduce the number of levels and adjust it again. The newly proposed mapping to the attack feasibility rating is shown in table 6.

Finally, in an attempt to address `UPDATE P4`, i.e., to have an option for a quicker attack feasibility rating decision, we propose the following: on a first iteration of the TARA, rather than doing the full attack potential calculation as outlined above, assign the levels based on required proximity: If the attack requires physical access assign the attack feasibility level of "Low" and if the attack can be done remotely, assign a level of "High". While obviously

Table 5: Attack feasibility parameter values

Expertise	Value	Knowledge of item	Value	Window of opportunity	Value	Equipment	Value
Multiple Experts	0	Critical	0	Small	0	Multiple bespoke	0
Expert	1	Sensitive	1	Medium	1	Bespoke	1
Proficient	2	Restricted	2	Large	2	Specialized	2
Layman	3	Public	3	Unlimited	3	Standard	3

Table 6: Attack feasibility rating calculation

Parameter Sum (A_{sum})	Attack feasibility rating (AF)
$0.00 \leq x < 0.30$	Very Low
$0.30 \leq x < 0.60$	Low
$0.60 \leq x < 0.80$	Medium
$0.80 \leq x \leq 1.00$	High

Table 7: Impact rating calculation

Parameter Sum (I_{sum})	Impact rating
$0.00 \leq x < 0.01$	Negligible
$0.01 \leq x < 0.05$	Moderate
$0.05 \leq x < 0.45$	Major
$0.45 \leq x \leq 1.00$	Severe

oversimplified, this allows a quick rating, so that more attention can be paid to the impact rating. This should allow to quickly dismiss some of the threat scenarios without spending a lot of time and effort on them without completely ignoring attack feasibility either.

4.2.4 Impact rating. As soon as the damage scenarios have been identified, the impact rating can be estimated. The four impact categories are safety, financial, operational and privacy. As noted in section 2, this requires a minor update, specifically `UPDATE G7`, to split out the legislative impact parameter from the privacy parameter. If still desired, legislation could be added as a separate impact category, extending the categories is explicitly allowed.

Another important update is `UPDATE G6`, the change in perspective to the road user as the primary stakeholder. In the presence of additional stakeholders, such as the OEM, it might make sense to add one or more new impact categories for additional stakeholders. For instance, a financial category for the OEM could be added in addition to the financial category for the road user.

For the impact rating calculation, which yields a single impact rating, the same updates apply as for the attack feasibility rating: an update to normalize the sum (`UPDATE P3`), and one to adjust the levels (`UPDATE G8`). In the published DIS of ISO/SAE 21434, a single impact rating is required in order to be able to use a risk matrix, but this is no longer the case in the latest draft of the standard, the use of risk matrices will be optional and no requirements are put on how to perform the impact rating. The normalization is analogous to the attack feasibility sum normalization, except that the parameter values use a logarithmic scale (0, 1, 10, 100) since increasing impact is exponentially worse (cf. table 15 in appendix B). So the equation becomes:

$$I_{nsum} = \frac{\sum_i^n w_i i_i}{100 * \sum_j^n (w_j)} \tag{5}$$

The new impact rating calculation which contains both updates is depicted in table 7, which is a rough translation of the original HEAVENS 1.0 table (cf. table 16 in appendix B).

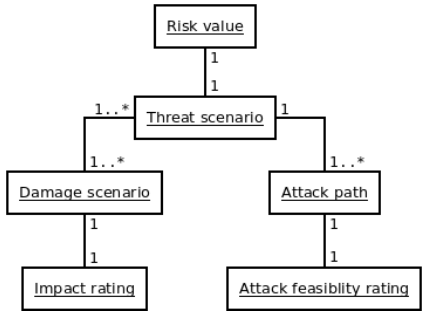


Figure 3: Threat scenario relationships and their multiplicities

4.2.5 Risk determination. The last step of the TARA focuses on determining the risk value. In order to understand this activity, it is important to clarify the relationships between risk value, threat scenario, damage scenario, impact rating, attack paths and attack feasibility rating. Figure 3 depicts these relationships and their multiplicities. Each threat scenario should have a single associated risk value, but a threat scenario can have multiple associated damage scenarios and attack paths which implies that there are multiple associated impact ratings and attack feasibility ratings.

ISO/SAE 21434 explicitly mentions the multiplicity problem for the attack feasibility rating and suggests to use the highest value assigned to any associated attack path, but the same is not acknowledged for damage scenarios. We recommend to use the highest value of both the associated impact ratings and attack paths, which ensures that the threat scenario is not underestimated.

As reflected in `UPDATE G10`, HEAVENS 1.0 only derived a security level, not a risk value, thus confounding the two. This is addressed by converting the matrix for determining the security level into a risk matrix. As noted by `UPDATE G9`, the number of levels need to be reduced as well. ISO/SAE 21434 specifies the use of five risk levels, but it does not mandate the use of a risk matrix. Table 8 shows the risk matrix we propose for HEAVENS 2.0. It is almost

Table 8: HEAVENS 2.0 – Risk matrix

		Impact rating			
		Negl.	Mod.	Maj.	Sev.
Attack feasibility rating	Very Low	1	1	2	3
	Low	1	2	3	4
	Medium	2	3	4	5
	High	2	4	5	5

symmetric, except for a lower risk value for negligible impact and high attack feasibility, since it seems odd to assign a medium risk value to a threat scenario with negligible impact. What constitutes acceptable risk is dependent on the concrete project, but we would argue that risk values of 1 or 2 represent acceptable levels of risk in most cases.

Once the initial risk determination is complete, and if the attack feasibility rating was approximated using only proximity, it might be beneficial to re-estimate the attack feasibility rating using the full process for the threat scenarios with the highest risk values. This can be done iteratively, focusing on the highest risks first.

4.2.6 Risk treatment decision. Once a threat scenario receives its risk value, a risk treatment decision has to be made, specifically if the risk should be avoided, shared or transferred, accepted, or reduced. Since this decision process was not explicitly included in HEAVENS 1.0, `UPDATE G11` demands its addition, which is already reflected in figure 2.

Risk avoidance can be achieved by removing the risk source or by stopping the activity which incorporates the risk. *Risk sharing or transferal* is typically achieved through contracts, for instance by taking out insurance against the risk, or by ensuring that an involved third party accepts a part of the risk. *Risk acceptance* means that the risk is deemed manageable without additional measures, and any assumptions which lead to the acceptability of the risk need to be documented. Finally, the most common option is *risk reduction*, which is typically achieved by including additional security controls in the item's architecture, in the item's production process, or in the implementing organization(s).

4.3 Cybersecurity claims and cybersecurity goals

In order to close the final gap to ISO/SAE 21434, `UPDATE G12` needs to be addressed, which is the addition of cybersecurity claims for accepted or transferred risks, and cybersecurity goals as an outcome of a planned risk reduction. *Cybersecurity claims* are statements about why a risk is acceptable and under which circumstances the acceptance decision needs to be re-evaluated. Cybersecurity claims also include assumptions which must be fulfilled for the risk to be acceptable, and claims must be documented for all accepted risks.

If a decision has been reached that a certain risk needs to be reduced, cybersecurity goals need to be defined for it. *Cybersecurity goals* are high-level cybersecurity requirements, which is equivalent to the high-level security requirements in HEAVENS 1.0. Moreover, ISO/SAE 21434 allows a cybersecurity goal to have a corresponding

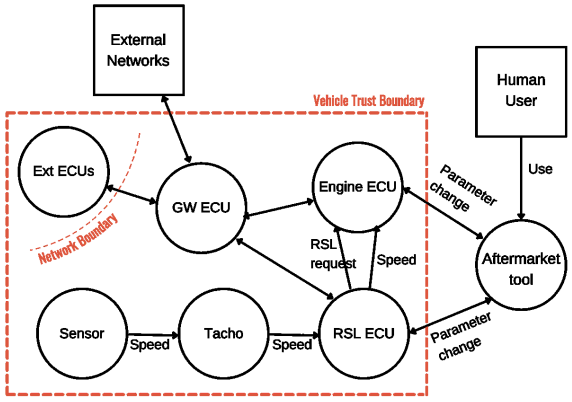


Figure 4: Road speed limit (RSL) item [3] data flow diagram

cybersecurity assurance level (CAL), which specifies a target level of process rigor for the validation processes.

After addressing all of the model updates, HEAVENS 2.0 fulfills all the requirements of the required risk assessment process in ISO/SAE 21434, as outlined in section 2. Despite its close ties to automotive use cases, after minor parameter calibrations HEAVENS 2.0 can also be applied to similar industries such as medical devices or industrial systems. This has been demonstrated in industrial applications which unfortunately are not available to the public.

5 EXAMPLE – THE SPEED LIMITER USE CASE

In order to demonstrate how certain aspects of the updated model HEAVENS 2.0 can be applied, we introduce a simple use case. For simplicity and to easily highlight the differences, we analyze the speed limiter example from the HEAVENS 1.0 paper [3].

Speed limiters are often used for commercial vehicles to guarantee that a driver can not exceed a set speed limit, which may even be required for regulatory compliance. Figure 4 depicts a data flow diagram (DFD) of the road speed limit (RSL) example. In this example, a sensor reports the current vehicle speed to a tachograph, which in turn reports it to a speed limiter electronic control unit (`RSL ECU`). The RSL ECU reports the vehicle speed to the engine ECU, and if necessary, the RSL ECU can request to lower the speed.

5.1 Item definition and asset identification

For this example, the data flow diagram (DFD) in figure 4 can function as the item definition and can also help with automatic asset identification.

The results of Tuma and Scandariato [29] indicate that a per-element STRIDE threat analysis is more effective than a per-interaction based one, so we focus on the elements as assets rather than the interactions. Therefore, all elements in the DFD are assets. In order to keep this example small, we will focus on a single element of the item, the engine ECU.

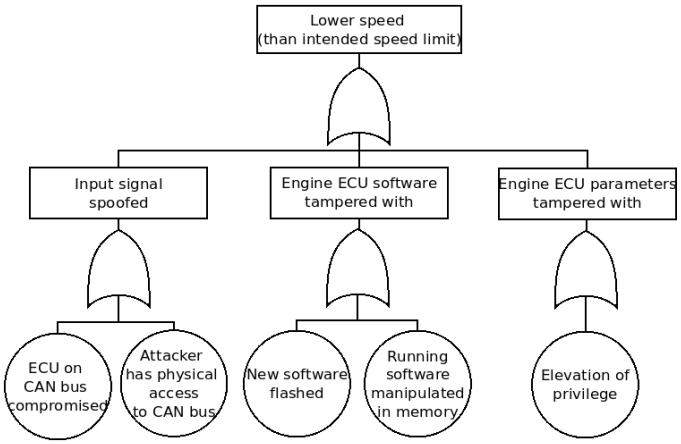


Figure 5: Attack tree for the engine ECU - damage scenario 'lower speed'

5.2 Threat scenario and damage scenario identification

In the context of the item, the following threat scenarios can be identified for the engine ECU: (1) **spoofing** of an input signal, (2) **tampering** with the software on the engine ECU to alter the reaction to an RSL request, (3) **elevation of privilege** to **tamper** with the engine ECU parameters to alter the reaction to an RSL request and (4) **denial of service**, where service is the reaction to RSL requests. Repudiation and information disclosure is not relevant for this particular element since logging would be handled on the RSL ECU and no sensitive information leaves the engine ECU.

Based on this threat analysis, we can identify the following damage scenarios as potential consequences of the threat scenarios: (a) the speed is lowered even though the speed limit has not been reached yet (potential consequence of threats 1, 2 and 3), or (b) the speed is not lowered, although the speed limit has been exceeded (potential consequence of threats 2, 3 and 4).

5.3 Attack path analysis, attack feasibility rating and impact rating

To save space, we will show only the attack path analysis of the first damage scenario, lowering the speed although the speed limit has not been reached yet. The corresponding attack tree is shown in figure 5. Each path from a leaf to the root constitutes an attack path, so there are five attack paths for this damage scenario. Note that most of the leafs could be extended further, but this level of detail should suffice to demonstrate the principle. Also note that the sub-trees for the threat scenarios "Engine ECU software/parameters tampered with" would be the same in the second damage scenario, a practical illustration that threat scenarios can lead to more than one damage scenario.

As this is the initial pass through the TARA, we will simplify the attack feasibility rating (AFR) to access means, i.e., physical (AFR = low) or remote (AFR = high). However, since we did not fully expand the attack paths, several of the attacks could be performed either physically or remotely, in which case we chose remote. In this particular case all attack paths are uniquely identified by a single leaf node. The resulting attack feasibility rating is shown in table 9.

Table 9: Attack feasibility rating for the attack paths of the "lower speed" damage scenario

ID	Attack Path Description	Attack feasibility rating (AF)
AP1	ECU compromised	High
AP2	Physical CAN access	Low
AP3	New software flashed	High
AP4	Manipulated software	High
AP5	Elevation of privilege	Low

The impact rating is calculated per damage scenario, so we only need to derive one impact rating for the case that the allowed speed is lower than the actually set speed limit. Assuming this is a commercial vehicle such as a truck, we want to add financial impact for the fleet owner (**fo**) as an impact category in addition to the four categories safety (**s**), financial (**f**), operational (**o**), and privacy (**p**) impact for the road user. We decide on a weight of 10 for both **s** and **fo** which leads to the following normalized formula:

$$I_{nsum} = \frac{10 * i_s + i_f + i_o + i_p + 10 * i_{fo}}{100 * (10 + 1 + 1 + 1 + 10)} \tag{6}$$

The safety impact for the road user will be low in the worst case ($i_s = 1$). Assuming the driver will not be blamed, financial losses will be absorbed by the fleet owner, so there is no direct financial impact for the driver ($i_f = 0$). The operational impact however is high since the vehicle can be inoperable in the worst case ($i_o = 100$). There is no privacy impact for the road user ($i_p = 0$), but the financial impact for the fleet owner might be medium (a significant financial loss that does not threaten bankruptcy) if this is an attack that scales remotely to large parts of the fleet ($i_{fo} = 10$):

$$I_{nsum} = \frac{10 * 1 + 0 + 100 + 0 + 10 * 10}{2300} = \frac{210}{2300} \approx 0.091 \tag{7}$$

According to table 7, the resulting impact rating is **Major**.

5.4 Risk determination, treatment decision and cybersecurity goals

When the attack feasibility ratings (AFRs) and the impact ratings (IRs) have been estimated, a risk value should be determined for each threat scenario. Since some of the threat scenarios have more than one attack path, we choose the attack paths with the highest AFR. Table 10 summarizes the result, including the resulting risk value for each threat scenario.

With the risk values being 5 and 3, there is no doubt that all three threat scenarios need to be mitigated, so the risk treatment decision is to reduce the risk. However, since no risks were low enough to be ignored, it may be prudent at this stage to return to the attack feasibility rating and do a full estimation instead of the access means approximation, in order to get a more realistic view. In this example, we will skip this step.

Table 10: Risk values for select threat scenarios of the engine ECU

Threat scenario	Attack feasibility rating (AFR)	Impact rating (IR)	Risk value
Input signal spoofed	High	Major	5
Software tampering	High	Major	5
Parameter tampering	Low	Major	3

This leaves only the formulation of cybersecurity goals: high-level requirements associated with threat scenarios. The following cybersecurity goals seem appropriate for the engine ECU: (1) *input signals to the engine ECU shall be authenticated*, (2) *input signals to the engine ECU shall be replay protected*, (3) *the engine ECU software shall be protected against tampering*, (4) *the engine ECU parameters shall be protected against unauthorized access*. Note that the formulations focus on what should be protected and are implementation independent. The goals will be broken down into detailed technical requirements in a later development stage. Since there was no shared, transferred or accepted risk, no cybersecurity claims have to be written.

6 STANDARDS, REGULATIONS AND RELATED WORK

The risk assessment processes which we covered in this paper are only a small part of ISO/SAE 21434: the standard includes requirements for the entire cybersecurity engineering process. For instance, the standard has three comprehensive clauses related to cybersecurity management and continuous cybersecurity activities. There are also clauses on all phases of the vehicle life-cycle, including product development, validation, production, operation, maintenance, and decommissioning. Finally, the standard also specifies that "cybersecurity interface agreements" should be reached with suppliers in order to clarify task responsibilities. In [11] Macher et al. reviewed ISO/SAE DIS 21434, and Macher and Schmittner also published an overview of automotive cybersecurity standards [26].

An important regulatory body for vehicles is the World Forum for Harmonization of Vehicle Regulations (WP.29) which is part of the United Nations Economic Commission for Europe (UNECE), and 54 countries have agreements to follow this regulatory body, including several non-European countries. By far the most important pieces of legislation recently passed on automotive cybersecurity are the UN vehicle regulations 155 and 156 (UNR 155 & 156), addenda to the 1958 agreement on vehicle harmonization by WP.29 [31, 32]. Among other requirements, UNR 155 mandates the use of documented risk assessment processes for cybersecurity, as well as the capabilities to detect and respond to attacks and to provide forensic capabilities. Moreover, the use of a cybersecurity management system (CSMS) is mandated for vehicle OEMs. UNR 156 requires the use of a "Software Update Management System (SUMS)" and also includes requirements for vehicle software to be updated and sufficiently secured against unauthorized access and updates. These regulations have been passed by WP.29 in June 2020 and have officially come into effect in January 2021, with plans for mandatory adoption in the various member states between 2021 and 2024 [30]. Moreover, ISO/SAE 21434 is referenced in UNR 155 as a possible source of appropriate risk assessment processes.

Since the original research on automotive system security by the EVITA project over a decade ago [18], many automotive risk assessment models building on these ideas have been proposed, a selection of which were reviewed by Macher et al. [9]. Like HEAVENS 1.0, many of these models predate ISO/SAE 21434 and do not fit directly into the risk assessment framework mandated by the standard. Some of them focus on the co-engineering of automotive safety and security [10, 12, 23–25, 27], while others focus on adapting security risk assessment models from other domains to the specifics of the automotive domain [3, 16, 21, 34], but a full description of these models with their advantages and disadvantages is out of scope for this paper.

However, two newer models have been proposed recently which do conform to ISO/SAE 21434, and we will shortly discuss similarities and differences with our work. In [22] Schmittner et al. explore the implications of UNR 155 [31], specifically of the requirement to introduce cybersecurity management systems (CSMS). They propose a high-level DevOps-based CSMS framework which can fulfill the requirements of both UNR 155 and ISO/SAE 21434. However, their CSMS framework is a high-level process description rather than a concrete risk assessment methodology, which is the main differentiating factor with our work.

The work by Wang et al. [33] is most closely related to HEAVENS 2.0. They also propose a risk assessment framework that seems to be based on HEAVENS 1.0 and is aligned with ISO/SAE 21434, but unlike our proposed HEAVENS 2.0 they make no efforts to address the practical shortcomings of HEAVENS 1.0 (cf. section 3). Additionally, they keep all the options given in the standard as options in their framework, so that their proposal is essentially the same as the risk assessment framework outlined in ISO/SAE 21434. For example, for the attack feasibility rating they propose to use one of three options: (1) an attack potential based on ISO/IEC 18045 [4], (2) the attack vector value from CVSS, or (3) a CVSS score. This is precisely the recommendation in the standard. In addition to being a nice tutorial to risk assessment in ISO/SAE 21434, their main contribution is the proposal of a new equation to determine the risk value, but they also propose an alternative risk matrix which is unexpectedly not aligned with the risk values required by ISO/SAE 21434.

Finally, modeling attack paths using attack graphs to estimate attack feasibility is in itself an active research area. An excellent overview and review of the different techniques was done by Kordy et al. [6]. Rather than trying to summarize the field, we refer to their work. In addition, recent publications apply these techniques to the automotive domain [8, 17], providing contemporary alternatives to the attack potential based approach presented in this paper.

7 CONCLUSION

Thanks to new legislation, ISO/SAE 21434 will see widespread adoption in industry and automotive companies need to learn how to integrate cybersecurity processes on project and organizational level. Threat analysis and risk assessment is one of the most prominent of these processes, and it seems especially prudent to apply

it in autonomous driving use cases to minimize the potential for maliciously caused safety incidents.

In order to facilitate the continued use of experiences from HEAVENS 1.0 in automotive projects, we analyzed its gap to the risk assessment framework mandated by ISO/SAE 21434. Consequently, we proposed 12 model updates to close this gap, and we also addressed 5 shortcomings identified for HEAVENS 1.0. Together, these 17 model updates form the basis for HEAVENS 2.0.

HEAVENS 2.0 functions as a drop-in threat analysis and risk assessment (TARA) model for ISO/SAE 21434. Practitioners who are already familiar with HEAVENS 1.0 will be able to learn this model easily and therefore be one step closer to applying ISO/SAE 21434. Finally, with minor parameter calibrations, HEAVENS 2.0 can also be applied to similar industries, such as medical devices or industrial systems.

ACKNOWLEDGMENTS

We would like to thank Christian Sandberg and Ulf Andersson for constructive discussions, and we would also like to thank all anonymous reviewers for their constructive feedback. The research leading to these results has been partially supported by VINNOVA, the Swedish Governmental Agency for Innovation Systems, through the project "Cyber Resilience for Vehicles - Cybersecurity for automotive systems in a changing environment (CyReV phase 2)" (2019-03071), and by the Swedish Civil Contingencies Agency (MSB) through the project "RICS2: Resilient Information and Control Systems".

REFERENCES

[1] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. 2011. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *Proceedings of the 20th USENIX Security Symposium*. San Francisco, CA, USA, 77–92.

[2] Mafijul Islam, Christian Sandberg, Andreas Bokesand, Tomas Olovsson, Henrik Broberg, Pierre Kleberger, Aljoscha Lautenbach, Anders Hansson, Andrew Söderberg-Rivkin, and Sathya Prakash Kadhirvelan. 2014. *Deliverable D2 - Security Models*. HEAVENS Project, Version 1.0 (Release 1).

[3] Mafijul Md. Islam, Aljoscha Lautenbach, Christian Sandberg, and Tomas Olovsson. 2016. A Risk Assessment Framework for Automotive Embedded Systems. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security* (Xi'an, China) *(CPSS '16)*. Association for Computing Machinery, New York, NY, USA, 3–14. https://doi.org/10.1145/2899015.2899018

[4] ISO/IEC. 2008. ISO/IEC 18045:2008 – Information technology – Security techniques – Methodology for IT security evaluation.

[5] ISO/SAE. 2021. ISO/SAE 21434 Road Vehicles - Cybersecurity Engineering.

[6] Barbara Kordy, Ludovic Piètre-Cambacédès, and Patrick Schweitzer. 2014. DAG-based attack and defense modeling: Don't miss the forest for the attack trees. *Computer Science Review* 13-14 (2014), 1–38. https://doi.org/10.1016/j.cosrev.2014.07.001

[7] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. 2010. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy (SP)*. IEEE, 447–462. https://doi.org/10.1109/SP.2010.34

[8] Stefano Longari, Andrea Cannizzo, Michele Carminati, and Stefano Zanero. 2019. A Secure-by-Design Framework for Automotive On-board Network Risk Analysis. In *2019 IEEE Vehicular Networking Conference (VNC)*. 1–8. https://doi.org/10.1109/VNC48660.2019.9062783

[9] Georg Macher, Eric Armengaud, Eugen Brenner, and Christian Kreiner. 2016. A Review of Threat Analysis and Risk Assessment Methods in the Automotive Context. In *Computer Safety, Reliability, and Security*, Amund Skavhaug, Jérémie Guiochet, and Friedemann Bitsch (Eds.). Springer International Publishing, Cham, 130–141.

[10] Georg Macher, Andrea Höller, Harald Sporer, Eric Armengaud, and Christian Kreiner. 2015. A Combined Safety-Hazards and Security-Threat Analysis Method for Automotive Systems. In *Computer Safety, Reliability, and Security*. Springer, 237–250.

[11] Georg Macher, Christoph Schmittner, Omar Veledar, and Eugen Brenner. 2020. ISO/SAE DIS 21434 Automotive Cybersecurity Standard - In a Nutshell. In *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, António Casimiro, Frank Ortmeier, Erwin Schoitsch, Friedemann Bitsch, and Pedro Ferreira (Eds.). Springer International Publishing, Cham, 123–135.

[12] Georg Macher, Harald Sporer, Reinhard Berlach, Eric Armengaud, and Christian Kreiner. 2015. SAHARA: A Security-Aware Hazard and Risk Analysis Method. In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition* (Grenoble, France) *(DATE '15)*. EDA Consortium, San Jose, CA, USA, 621–624.

[13] CCRA members. 2017. Common Methodology for Information Technology Security Evaluation.

[14] Charlie Miller and Chris Valasek. 2013. *Adventures in automotive networks and control units*. Retrieved October 25, 2020 from http://illmatics.com/car_hacking.pdf

[15] Charlie Miller and Chris Valasek. 2015. *Remote Exploitation of an unaltered Passenger Vehicle*. Retrieved October 25, 2020 from http://illmatics.com/Remote%20Car%20Hacking.pdf

[16] Jean-Philippe Monteuuis, Aymen Boudguiga, Jun Zhang, Houda Labiod, Alain Servel, and Pascal Urien. 2018. SARA: Security Automotive Risk Analysis Method. In *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security* (Incheon, Republic of Korea) *(CPSS '18)*. Association for Computing Machinery, New York, NY, USA, 3–14. https://doi.org/10.1145/3198458.3198465

[17] Christian Plappert, Daniel Zelle, Henry Gadacz, Roland Rieke, Dirk Scheuermann, and Christoph Krauß. 2021. Attack Surface Assessment for Cybersecurity Engineering in the Automotive Domain. In *2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. 266–275. https://doi.org/10.1109/PDP52278.2021.00050

[18] Alastair Ruddle, David Ward, Benjamin Weyl, Sabir Idrees, Yves Roudier, Michael Friedewald, Timo Leimbach, Andreas Fuchs, Sigrid Gürgens, Olaf Henniger, Roland Rieke, Matthias Ritscher, Henrik Broberg, Ludovic Apvrille, Renaud Pacalet, and Gabriel Pedroza. 2009. *Security requirements for automotive on-board networks based on dark-side scenarios*. EVITA Project, Deliverable D2.3, v1.1.

[19] SAE International. 2016. SAE J3061_201601 - Cybersecurity Guidebook for Cyber-Physical Vehicle Systems.

[20] Christian Sandberg, Andreas Bokesand, and Urban Thorsson. 2018. *HoliSec Deliverable D4.1.1 - Tailoring the HEAVENS risk assessment methodology for improved performance*. Technical Report.

[21] Karsten Schmidt, Peter Tröger, Hans-Martin Kroll, Thomas Bünger, Florian Krueger, and Christian Neuhaus. 2014. Adapted Development Process for Security in Networked Automotive Systems. *SAE International Journal of Passenger Cars - Electronic and Electrical Systems* 7, 2 (2014), 516–526. https://doi.org/10.4271/2014-01-0334 arXiv:http://saepcelec.saejournals.org/content/7/2/516.full.pdf+html

[22] Christoph Schmittner, Jrgen Dobaj, Georg Macher, and Eugen Brenner. 2020. A Preliminary View on Automotive Cyber Security Management Systems. In *Proceedings of the 23rd Conference on Design, Automation and Test in Europe* (Grenoble, France) *(DATE '20)*. EDA Consortium, San Jose, CA, USA, 1634–1639.

[23] Christoph Schmittner and Zhendong Ma. 2015. Towards a Framework for Alignment Between Automotive Safety and Security Standards. In *Computer Safety, Reliability, and Security*, Floor Koornneef and Coen van Gulijk (Eds.). Lecture Notes in Computer Science, Vol. 9338. Springer International Publishing, 133–143. http://dx.doi.org/10.1007/978-3-319-24249-1_12

[24] Christoph Schmittner, Zhendong Ma, and Erwin Schoitsch. 2015. Combined safety and security development lifecycle. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. IEEE, 1408–1415. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7281940

[25] Christoph Schmittner, Zhendong Ma, Erwin Schoitsch, and Thomas Gruber. 2015. A Case Study of FMVEA and CHASSIS as Safety and Security Co-Analysis Method for Automotive Cyber-Physical Systems. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security* (Singapore, Republic of Singapore) *(CPSS '15)*. Association for Computing Machinery, New York, NY, USA, 69–80. https://doi.org/10.1145/2732198.2732204

[26] Christoph Schmittner and Georg Macher. 2019. Automotive Cybersecurity Standards - Relation and Overview. In *Computer Safety, Reliability, and Security*, Alexander Romanovsky, Elena Troubitsyna, Ilir Gashi, Erwin Schoitsch, and Friedemann Bitsch (Eds.). Springer International Publishing, Cham, 153–165.

[27] Erwin Schoitsch, Christoph Schmittner, Zhendong Ma, and Thomas Gruber. 2016. The Need for Safety and Cyber-Security Co-engineering and Standardization for Highly Automated Automotive Vehicles. In *Advanced Microsystems for Automotive Applications 2015*, Tim Schulze, Beate Müller, and Gereon Meyer (Eds.). Springer International Publishing, Cham, 251–261. https://doi.org/10.1007/978-3-319-20855-8_20

[28] Adam Shostack. 2014. *Threat modeling – designing for security*. Wiley.

[29] Katja Tuma and Riccardo Scandariato. 2018. Two Architectural Threat Analysis Techniques Compared. In *Software Architecture (Lecture Notes in Computer*

Science), Carlos E. Cuesta, David Garlan, and Jennifer Pérez (Eds.). Springer International Publishing, Cham, 347–363.

[30] UNECE. 2020. *UN Regulations on Cybersecurity and Software Updates to pave the way for mass roll out of connected vehicles.* Retrieved October 25, 2020 from http://www.unece.org/?id=54667

[31] UNECE. 2021. *Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system.* Retrieved March 25, 2021 from https://unece.org/sites/default/files/2021-03/R155e.pdf

[32] UNECE. 2021. *Uniform provisions concerning the approval of vehicles with regards to software update and software updates management system.* Retrieved March 25, 2021 from https://unece.org/sites/default/files/2021-03/R156e.pdf

[33] Yunpeng Wang, Yinghui Wang, Hongmao Qin, Haojie Ji, Yanan Zhang, and Jian Wang. 2021. A Systematic Risk Assessment Framework of Automotive Cybersecurity. *Automotive Innovation* (2021), 1–9.

[34] Marko Wolf and Michael Scheibel. 2012. A systematic approach to a qualified security risk analysis for vehicular IT systems. In *Automotive - Safety & Security 2012 (Lecture Notes in Informatics)*, Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, and Silke Spitzer (Eds.). Gesellschaft für Informatik, Bonn, 195–210.

[35] Marko Wolf, André Weimerskirch, and Christof Paar. 2004. Security in Automotive Bus Systems. In *Workshop on Embedded IT-Security in Cars*. Bochum, Germany.

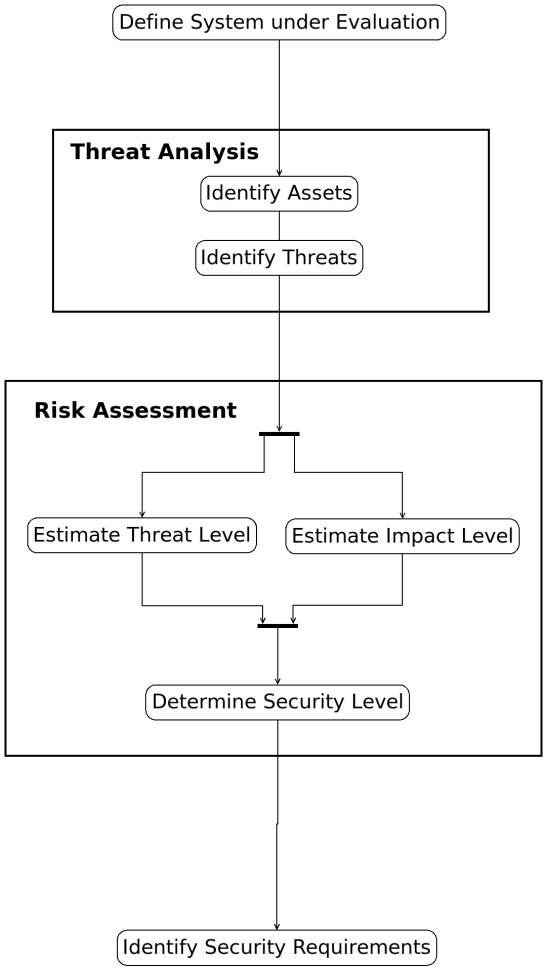


Figure 6: HEAVENS 1.0 workflow

A SUPPORTING MATERIAL

Table 11 shows a summary of the proposed updates that lead to HEAVENS 2.0, thus highlighting the differences to HEAVENS 1.0.

B HEAVENS 1.0

For completeness, we briefly introduce the workflow and key concepts of the HEAVENS 1.0 security model in this appendix, including all important tables and formulas as well as parameter descriptions, reproduced with permission from the authors [3]. The workflow is depicted in figure 6. It starts with a basic definition of the system being evaluated. This is followed by the threat analysis phase which has two parts: identifying the assets and identifying the threats. In HEAVENS 1.0 this is done by using the STRIDE model, a simple keyword based guidance technique, on a data flow diagram (DFD). By creating a DFD, the assets are implicitly identified: all entities and data flows in the diagram are assets, and depending on the type of asset, certain threats from STRIDE apply automatically. This allows an (automatic) enumeration of all asset/threat pairs.

Every asset/threat pair is then fed into the risk assessment process. Traditional risk assessment follows the formula of $risk = likelihood * impact$, and the HEAVENS 1.0 risk assessment process approximates this formula as well. The process has three main parts: estimating the threat level (likelihood), estimating the impact level, and finally determining a security level based on these estimates.

The threat level estimation is based on an adapted attack potential calculation from the vulnerability analysis methodology in appendix B.4 of ISO/IEC 18045 [4]. The underlying idea is to evaluate several parameters which together form an estimate of the likelihood that a particular threat will be exploited. The threat level parameters in HEAVENS 1.0 are: "Expertise", "Knowledge about target", "Window of opportunity" and "Equipment" (see table 13). Every asset/threat pair is rated for each of these parameters according to predefined levels, and the result is combined to form a single threat level rating (table 14).

The impact level estimation works on a similar premise, namely that the impact can be evaluated separately for four impact categories which are then combined to form a single impact level. These four impact categories are: "Safety", "Financial", "Operational" and "Privacy/Legislative" (table 15). Since not every impact category might be equally important, they can be weighted. The default recommendation in HEAVENS 1.0 is to weight the "Safety" and "Financial" categories as 10 times more important than the "Operational" and "Privacy/Legislative" impact, as also proposed by Wolf and Scheibel [34], but the weights can be adjusted as needed.

Once the threat level and impact level have been estimated, a corresponding security level can be determined for the asset/threat pair. For this purpose, a "risk matrix" has been defined, similar to the way automotive safety integrity levels (ASILs) are determined in ISO 26262. This matrix is shown in table 17. HEAVENS 1.0 defines five security levels: "quality management (QM)", "low", "medium", "high" and "critical". "QM" simply means that traditional quality management processes are sufficient and no additional security requirements need to be defined for this threat.

Finally, if the security level is determined to be above "QM", a high-level (implementation independent) security requirement to mitigate the threat should be defined. Ultimately, the security level corresponds to a risk level of the threat. It follows a listing of the tables and parameter descriptions of HEAVENS 1.0.

STRIDE. The STRIDE mnemonic is presented in table 12.

Threat level parameters. The threat level parameters are shown in table 13, and a summary of the parameters follows.

Expertise. The general level of knowledge required to carry out an attack:

- Layman. No particular expertise is required.
- Proficient. General security and domain knowledge is required. Professionals with knowledge about simple and popular attacks, are capable of mounting them with available tools, and if necessary, are able to improvise.
- Expert. Expert security and domain knowledge is required. Experts are familiar with underlying algorithms, protocols, hardware, software and concepts. They know techniques and tools of existing attacks and are able to create new attacks.
- Multiple Experts. Expert security and domain knowledge is required for several distinct domains. Allows for a situation

Table 11: Summary of proposed model updates ordered by workflow activities

Update	Activity	Description
UPDATE G1	-	Align terminology
UPDATE G2	-	Merge threat analysis and risk assessment phases
UPDATE G3	Damage scenario identification	Include damage scenario identification
UPDATE G4	Attack path analysis	Include attack path analysis
UPDATE G5	Attack feasibility rating	Adjust threat levels
UPDATE P1	Attack feasibility rating	Inverse threat level sum
UPDATE P2	Attack feasibility rating	Normalize threat level parameter sum
UPDATE P4	Attack feasibility rating	Simplify threat level estimation for threat pre-screening
UPDATE P5	Attack feasibility rating	Add sub-parameters for "window of opportunity"
UPDATE G6	Impact rating	Shift the stakeholder perspective
UPDATE G7	Impact rating	Remove legislative impact parameter
UPDATE G8	Impact rating	Adjust impact levels
UPDATE P3	Impact rating	Normalize impact level parameter sum
UPDATE G9	Risk determination	Rename security level to risk value
UPDATE G10	Risk determination	Adjust security levels
UPDATE G11	Risk treatment decision	Include risk treatment decision and resulting actions
UPDATE G12	Risk treatment decision	Include cybersecurity claims and goals

Table 12: Microsoft's STRIDE methodology [28]

Threat	Violated Attribute	Explanation
Spoofing	Authenticity	Attackers pretend to be someone or something else
Tampering	Integrity	Attackers change data in transit or in a data store
Repudiation	Non-repudiation	Attackers perform actions that cannot be traced back to them
Information disclosure	Confidentiality/Privacy	Attackers get access to data (e.g. in transit or in a data store)
Denial of Service	Availability	Attackers interrupt a system's legitimate operation
Elevation of privilege	Authorisation	Attackers perform actions they are not authorised to perform

Table 13: Threat level parameter values

Expertise	Value	Knowledge about target	Value	Window of opportunity	Value	Equipment	Value
Layman	0	Public	0	Unlimited	0	Standard	0
Proficient	1	Restricted	1	Large	1	Specialised	1
Expert	2	Sensitive	2	Medium	2	Bespoke	2
Multiple Experts	3	Critical	3	Small	3	Multiple bespoke	3

Table 14: Threat level calculation

Parameter Sum (T_{sum})	Threat Level (TL)	TL Value
$10 - 12$	None	0
$7 - 9$	Low	1
$4 - 6$	Medium	2
$2 - 3$	High	3
$0 - 1$	Critical	4

Table 15: Impact level parameter values

Safety	Operational	Financial	Privacy and Legislative	Value
None	None	None	None	0
Low	Low	Low	Low	1
Medium	Medium	Medium	Medium	10
High	High	High	High	100

Table 16: Impact level calculation

Parameter Sum (I_{sum})	Impact Level (IL)	IL Value
0	None	0
$1 - 19$	Low	1
$20 - 99$	Medium	2
$100 - 999$	High	3
≥ 1000	Critical	4

Table 17: Calculation of security level from impact and threat level

Security Level (SL)		Impact Level (IL)				
		0	1	2	3	4
Threat Level (TL)	0	QM	QM	QM	QM	Low
	1	QM	Low	Low	Low	Medium
	2	QM	Low	Medium	Medium	High
	3	QM	Low	Medium	High	High
	4	Low	Medium	High	High	Critical

in which different fields of expertise are required at an expert level to succeed with an attack.

Knowledge about target. The distribution of information about the target, i.e., the availability of information and the community size possessing that knowledge. This parameter points to the sources from where attackers can gain knowledge about the target and indicates how difficult it is for an attacker to acquire that knowledge:

- Public. The necessary information is public.
- Restricted. The information is shared with partners under non-disclosure agreements.
- Sensitive. The information is shared between specific teams, but access is constrained to their members.
- Critical. The information is restricted to a few individuals. Access is tightly controlled on a strict need to know basis.

The first two levels, "Public" and "Restricted", specify knowledge distribution outside a single organization, whereas "Sensitive" and "Critical" specify knowledge distribution within a single organization. The attack potential decreases from "Public" to "Critical" due to the increasing difficulty for an attacker to obtain necessary information about the target.

Window of opportunity. The access type available to the attacker, and the time window the attacker has to mount a successful attack. The access type can be remote or physical:

- Unlimited. Unlimited physical access, or network access for an unlimited time.
- Large. High physical and/or remote availability with some time limitations.
- Medium. Low availability with severe time limitations. Limited physical and/or remote access to the target. Physical access to the vehicle interior or exterior without using any special tools.
- Small. Very low availability. Physical access required to perform complex disassembly of vehicle parts to access internals to mount an attack on the asset.

Equipment. is the equipment required to identify or exploit vulnerabilities. This can be hardware or software:

- Standard. The equipment is readily available to the attacker. The equipment may be part of the target itself (e.g. a debugger in an operating system), or is easily obtained.
- Specialized. The equipment is not readily available to the attacker, but could be acquired without undue effort. This could include the purchase of moderate amounts of equipment, or the development of more extensive attack scripts.
- Bespoke. The equipment is not readily available to the public as it may need to be specially produced, or because the equipment is so specialized that its distribution is controlled

or restricted. Alternatively, the equipment may be very expensive. Multiple types of specialized equipment required for a successful attack also fall under this category.
- Multiple Bespoke. Multiple types of bespoke equipment are required for a successful attack.

Threat level sum. The weighted threat level parameter sum is calculated as follows:

$$T_{sum} = w_x t_x + w_k t_k + w_w t_w + w_e t_e \tag{8}$$

where w_i and t_i are the weight and the estimated threat level value of parameter i, and the indices x, k, w, e stand for the four parameters, respectively. We assume that the parameters are of equal importance, i.e., $w_i = 1\ \forall i$, so the equation is simplified to:

$$T_{sum} = t_x + t_k + t_w + t_e \tag{9}$$

Impact level parameters. The impact level parameters are shown in table 15, and a summary of the parameters follows.

Financial impact.

- No impact. No discernible effects or appreciable consequences for the stakeholders.
- Low impact. The financial damage remains tolerable for the stakeholders.
- Medium impact. There are substantial financial losses which do not threaten the existence of the stakeholders.
- High impact. The financial damage threatens the existence of the stakeholders.

Operational impact. refers to operational damages, for instance the loss of secondary functionalities such as cruise control, or comfort and entertainment systems such as a CD-player or air-conditioning.

- No impact. No discernible effect.
- Low impact. The appearance of an item or an audible noise annoys between 25% and 75% of customers.
- Medium impact. Corresponds to the degradation or loss of a secondary function, or the degradation of a primary function.
- High impact. Corresponds to the loss of a primary function which leaves the vehicle inoperable and potentially affects safety or legislative aspects.

Privacy and Legislative impact. Deals with damages caused by privacy violations of stakeholders or violations of governmental regulations such as environmental or traffic laws.

- No impact. No discernible effect.
- Low impact. Corresponds to privacy violations without direct potential for abuse, or legislative violations with no appreciable consequences, e.g., a warning without a fine.
- Medium impact. Corresponds to privacy violations which lead to abuse, or legislative violations with business and financial impact such as fines or reputation loss.
- High impact. Corresponds to privacy violations of multiple stakeholders which lead to abuse, or legislative violations with significant business and financial impact, such as significant loss of market share, trust or reputation.

Impact level sum. The weighted impact level parameter sum is calculated as follows:

$$I_{sum} = w_s i_s + w_f i_f + w_o i_o + w_p i_p \tag{10}$$

where w_j and i_j are the weight and the estimated impact value of parameter j, and the indices s, f, o, p stand for the parameters Safety, Financial, Operational and Privacy and Legislative, respectively. With the default weights, this simplifies to:

$$I_{sum} = 10\ (i_s + i_f) + i_o + i_p \tag{11}$$

Security level. Finally, the security level is derived using the matrix in table 17.