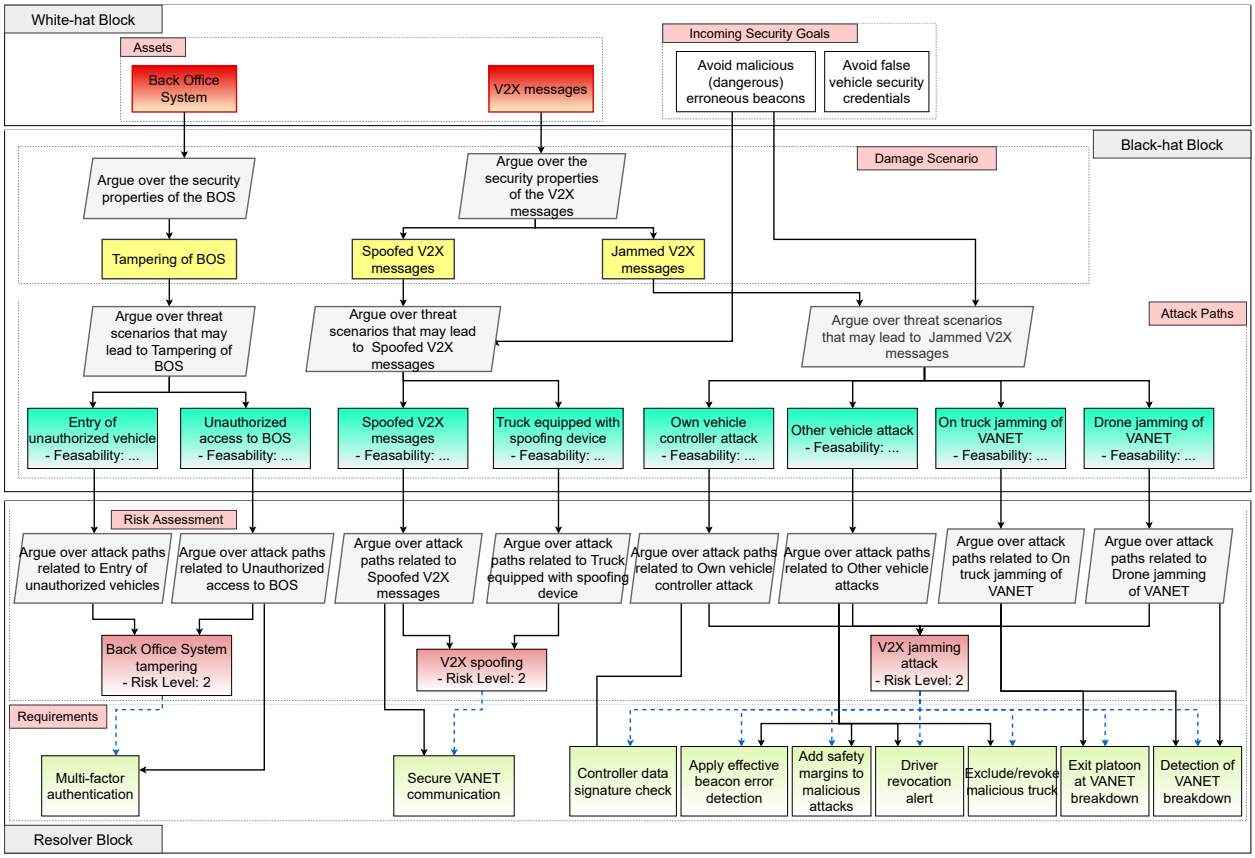
CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG



Supporting the Creation of Security Assurance Cases for Automotive Companies

Master's thesis in Software Engineering and Technology

Sarosh Jah Nasir & Vamsi Ravi

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Master's thesis 2021

Supporting the Creation of Security Assurance Cases for Automotive Companies

Sarosh Jah Nasir & Vamsi Ravi

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2021

Supporting the Creation and Maintenance of Security Assurance Cases for Automotive Companies

Sarosh Jah Nasir & Vamsi Ravi

© Sarosh Jah Nasir & Vamsi Ravi, 2021.

Supervisors: Jan-Philipp Steghöfer, Department of Computer Science & Engineering and Mazen Mohamad, Department of Computer Science & Engineering
Advisor: Ali Shahrokni, Systemite AB
Examiner: Christian Berger, Department of Computer Science & Engineering

Master's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2021

Supporting the Creation of Security Assurance Cases for Automotive Companies
Sarosh Jah Nasir & Vamsi Ravi
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Security Assurance Cases (SACs) have gained significant focus in recent years, especially in safety-critical industries such as the automotive industry. Furthermore, there has been a push towards connected cars technology in vehicles, which means that vehicles to a greater extent are exposed to cyber threats. Because of this, a new standard, ISO/SAE-21434, is currently under development, which requires automotive companies to start using SACs to ensure the security of their vehicles against cyber attacks. Using the Design Science Research (DSR) method, two iterations are conducted in which the first iteration focuses on identifying artifacts from Automotive Development Processes (ADPs) that could be used in the creation of SACs. The second iteration investigates to what extent the identified artifacts cover the needs of the approaches suggested in literature. Two open source catalogues are created as the artifact of the first iteration. The second iteration is a gap analysis, including the creation of two SACs, and a a SAC Report Template. The catalogues are used as an aid during the creation of the SACs, as well as quality assurance to assess the quality of the cases. The identified gaps are presented, discussed, and validated by the case company and a third party. The catalogues and the SAC Report Template were implemented into SystemWeaver, a system engineering tool manufactured by the case company Systemite. The artifacts created from this thesis can be used in the future to support practitioners in the creation of SACs.

Keywords: Security, Assurance Cases, Security Case, Automotive Industry, Cybersecurity, ISO/SAE-21434, SystemWeaver.

Acknowledgements

We would like to give a big heartfelt thank you to our supervisors Mazen Mohamad and Jan-Philipp Steghöfer for their continuous support, guidance and feedback. Special thanks to Ali Shahrokni, Jan Söderberg, and the rest of the Systemite team for collaborating with us to make this thesis happen.

Sarosh Jah Nasir & Vamsi Ravi, Gothenburg, September 2021

Contents

List of Figures

1
Introduction

Modern day vehicles are becoming more software based as we move on from traditional mechanical connections in vehicles to using more computer-controlled systems with various sensors to introduce systems such as adaptive cruise control and self-steering [1]. This progress in the automotive industry has been on-going for a couple of decades now to the point where we have gone from systems working offline (private networks) to now having connected cars that rely on real-time networking, becoming a part of the Internet of Things and providing different types of services, such as infotainment and smartphone integration [2]. As a consequence of this development, each vehicle is now more exposed to cyber threats and there is an increased risk of cyber attacks. [3]. Furthermore, there has not been as much progress into making assurance cases for claims about security, particularly in regards to cybersecurity, in this domain. Although there is a lack of using Security Assurance Cases (SACs) in the industry, there are papers about how to create SACs in different domains, including the automotive domain [4]. Additionally, new standards are being made to aid the progress of creating Security Assurance Cases in such domains.

Currently, there has only been investigative work on how to create SACs within the automotive industry, but no actual methodology is being practiced. With the new standards ISO-26262 Road vehicles - Functional safety [5] and ISO/SAE-21434 Road vehicles - Cybersecurity engineering (under development) [6], the need for Security Assurance Cases is on its way to becoming an essential part of Automotive Development Processes (ADPs), as it is a way for validating (or invalidating) security claims about automotive vehicles. The UNECE [7] has a section on cybersecurity principles where they have defined a set of principles that vehicle manufacturers should follow. For example, statement 3.3.9 in UNECE states that vehicles should be resilient against cyberattacks. The industry, however, is still in a phase of adopting SACs [4].

Assurance cases are structured bodies of evidence that support claims about systems with the use of arguments. An assurance case includes important information about the case, such as claims, arguments, and evidence, just to name a few [8]. The cases take a top-down approach where a claim about a system is set and is supported by objective arguments and evidence to support the claim, not to mention that top level claims can also be supported by sub-claims, depending on the chosen strategy. The larger the system for the automotive vehicle, the more elaborate the SAC needs to be, as there are more automotive assets to consider. Security Assurance Cases are specialized in cyber security, where the claims are about the security of a system [4].

This study has been conducted in partnership with the case company Systemite, which will henceforth be referred to as 'the case company'. The case company are the manufacturers of the development platform SystemWeaver [9] used within the automotive industry. This platform provides users with a single system engineering tool for all steps of the development process, including: requirements, architecture design, component design, testing, and more. Through the collaboration with the case company, we have access to practitioners in the automotive domain that can provide valuable data during this study.

1.1 Problem Domain & Motivation

Literature within the field of SACs provides suggestions to approaches for developing SACs, however nothing concrete has emerged for the Automotive Development Process. The case company, Systemite, has conducted investigations on how to represent SACs, but there is still a need for further knowledge about artifacts that are created during an ADP and whether these artifacts would be enough for the creation of a SAC. An artifact found in a ADP, in this context, is something in an automotive system that can be used to create arguments and evidence of SACs. In addition to the deficient knowledge about this topic, there is also no online documentation of these artifacts. The lack of such catalogues or documentation prevents practitioners from focusing solely on creating SACs. Instead, they must first take a step into preparing and gathering the necessary data before they can start the SAC creation process.

Another problem when it comes to the creation of SACs is the gap between research and industry. Research has presented many suggestions to approaches in creating SACs, however, these have yet to be adopted in the industry. This is evident by the amount of papers that exist about SACs in proportion to the usage of SACs by the automotive industry [4].

Our case company has raised similar issues as mentioned above in regards to the challenges that they are facing today with their system engineering tool, SystemWeaver. Thus, this is another reason for our research into supporting the creation of Security Assurance Cases in the automotive industry.

1.2 Research Goal & Research Questions

The goal of this study is to assist practitioners in creating Security Assurance Cases. For this purpose, we have brought forth a couple of research questions that will help reach our goal.

RQ1. Which artifacts are created during an Automotive Development Process that can be used in Security Assurance Cases?
   1. What are the assets, vulnerabilities, and controls which can support the creation of SAC arguments?

2. What evidence exists that can support the claim of the arguments?

RQ2. To what extent do these artifacts cover the needs in the approaches suggested in literature?

The first research question aims to provide a catalogue for practitioners in the field in order to assist the creation of SACs. The catalogue will not only consist of assets, vulnerabilities, and controls (mitigation strategies) but also existing evidence that can support the claims in a SAC.

For the second research question, an investigation will be done on the gap between literature and industry. In particular, what approaches researchers are suggesting in order to create SACs and to what extent the Automotive Development Process covers the needs of the suggested approaches. With new standards coming out, this investigation provides an insight into whether the approaches suggested today need to be modified to fulfill the requirements set by the new standards.

1.3 Contribution & Scope

The provided artifacts include two catalogues; one with assets, vulnerabilities, and controls, and the other with types of evidence, that are within an ADP and can be used to support both practitioners and companies in developing security cases. The catalogues have been integrated with the system engineering tool manufactured by our case company. Additionally, we have provided links for the catalogue to be used as open source catalogues for people who are interested in them.

The gap analysis that was conducted provides insight to the automotive industry about the current state of SACs, to some extent. It includes our experience of creating SACs using a specific approach. In addition to that, the SAC report template is a suggestion of how the case company, as well as other companies could structure and document their SACs, or simply derive their own report templates from ours, when they have started to adopt the creation of SACs.

The scope of this thesis is mainly within the automotive industry. For companies that have yet to adopt the creation of SACs to companies that have started that process. However, we can see these contributions being used in other domains that may be involved in cybersecurity, as we have seen SACs being created for the medical domain [10].

1.4 Structure

This paper starts with a brief look into the introduction then background about the domain and thesis topic, followed by relevant related work. It then moves on towards the section where the chosen methodology is described and provides information about what was done in each part of the methodology. Then it presents the results of the research including some thoughts about the findings, which it further delves

into during the analysis and discussion section of the paper. Finally, conclusions and final thoughts about the results are presented, along with a look into the future work that is needed regarding this topic.

2
Background

In order to better understand what this research entails, a few things should be explained. ECUs, ISO Standards, and assurance cases have been introduced. This section aims to provide further insight on those topics, as well as other terms.

2.1 Systemite

Systemite is a Gothenburg, Sweden based company that has obtained an important position in the automotive industry, as well as telecom and military systems. Systemite has manufactured the first high performance open platform, SystemWeaver, for component-based systems development within the area of computer based systems [11]. SystemWeaver allows their customers to work on their development process in a single tool, where otherwise they might have been using different interfaces; for the analysis, design, architecture, etc [9].

Some of the steps that SystemWeaver supports are: Requirements, Architecture Design, and Test. What this entails is that you can work with requirements from different levels, including complete product requirements down to detailed design requirements, while staying fully integrated with other processes (requirement authoring, impact analysis, traceability, and more). In terms of architecture design, you can configure the architecture support needed with regards to different aspects within architecture design. As for tests, SystemWeaver covers specification of test cases, test planning, and execution. Solutions are fully integrated with all requirements models including requirement traceability. It's a tool that is designed to fit their customers needs by letting ten or a thousand people work together, allowing them to keep track of changes and control, adapt the system to suit their needs, and integrate SystemWeaver with external applications easily with its open API [9].

2.2 ISO Standards

The International Organization for Standardization (ISO) is an independent, non-governmental internal organization that develops International Standards that support innovation and provide solutions to global challenges [12]. There are over 23,500 internal standards that have covered various aspects of technology and manufacturing, including road vehicles. These ISO standards have been developed voluntarily by experts internationally. Essentially, they are a guide, how-to, or formula that best describe a way of doing something. Some categories they cover are Quality Management, Energy Management, and IT Security. Among those standards are two that

are of interest to the automotive industry: ISO-26262 Road Vehicles - Functional safety [5] and ISO/SAE-21434 Road Vehicles - Cybersecurity engineering [6]. As the title states, ISO-26262 focuses on the functional safety of a vehicle, specifically the safety of automotive electrical/electronic systems. On the other hand, ISO/SAE-21434 is about cybersecurity, i.e., standards to prevent cyberattacks against the systems in an automotive vehicle. This could include the system that controls the breaks for example. The latter standard is the one that is of most important to this thesis as it works closely with creating Security Assurance Cases.

Apart from the ISO/SAE-21434 standard, there is another form of document/standard by the UNECE. The Draft Recommendation on Cyber Security of the Task Force on Cyber Security and Over-the-air issues of UNECE WP.29 GRVA [7] is a document developed based on a number of different standards including the two ISO standards mentioned earlier. It's a handy document that includes useful information when developing an automotive vehicle, such as cyber security principles, threats to vehicle systems and mitigations. In order for companies to comply with these standards and provide assurance that they follow the standards, assurance cases are used.

2.3 Assurance Cases

The definition of an assurance case, as described in the Goal Structuring Notation (GSN) Community Standard V.2 [13], is as follows:

"A reasoned and compelling argument, supported by a body of evidence, that a system, service or organisation will operate as intended for a defined application in a defined environment."

Assurance cases are a collection of evidence brought forth in structured arguments. They are used to argue that a particular claim about a system's property holds [14]. In the case of a Security Assurance Case (SAC), the claims are about the security aspect of a system where the evidence is used to strengthen the aforementioned claim. SACs can be created with the help of Goal Structuring Notation (GSN), which "represents the individual elements of any safety argument [...] and (perhaps more significantly) the relationships that exist between these elements" [15]. The GSN provides the types of nodes that are needed in order to create a case [16].

1. Claim - The claim is the statement that the Security Assurance Case attempts to prove.
2. Context - The context refers to the top-level claim. It can be an item boundary or the systems architecture.
3. Strategy - The strategy is the method used to provide arguments to support the claim(s).
4. Assumption - Assumptions are made to strategies to infer certain properties about a claim.
5. Evidence - The proof that verifies the stated claim.

The case consists of a top-level claim which is broken down into sub-claims based on certain strategies. The claims are used to specify the goals of the case, e.g., a

certain system/feature is secure [4]. A strategy, for instance, can be broken down based on certain security attributes. The claims are broken down repeatedly till they reach a point where evidence can be assigned to justify the claims/sub-claims. Examples of evidence can be code review reports and/or test results. Assumptions can be made while applying the strategies.

Figure 2.1 displays the GSN nodes and example structure of an SAC [17]. Here, we can see the different nodes and identify them by their shapes. Claims are rectangular in shape, strategies are parallelograms, contexts are rounded rectangles, assumptions are oval and evidence are circles. In the example structure shown, "Product is adequately secure" is the top-level claim for which we have also provided a context. Our strategy to prove this claim is to "Argue per security requirement", given that "All security requirements have identified" (assumption). Following this strategy, we get two sub-claims: "Access control requirements are fulfilled" and "Private data is not disclosed". The nodes mentioned up till now are the argumentation for the top-level claim and lead to the evidence, that have provided "Test result of access control" for the top sub-claim, and "Results of information flow analysis" for the other sub-claim.

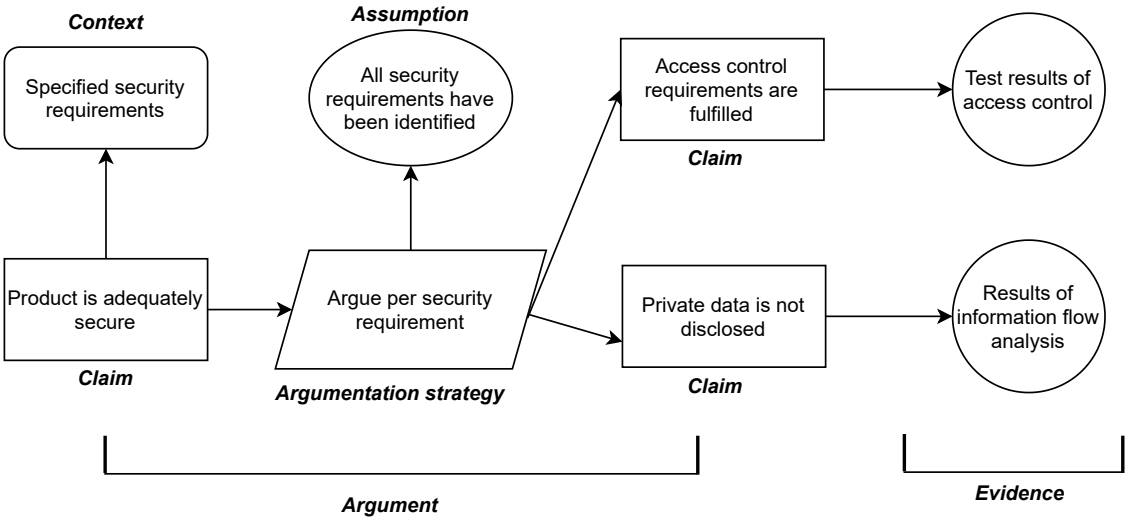


Figure 2.1: An example structure of an SAC using GSN. A caption is provided for each node but they are also identifiable by their shape.

There are many different ways to create SACs. In a previous study of SACs by Mohamad et al [4], the study found that there are around 26 different approaches. Approaches, sometimes called patterns, are methods of creating an SAC and differ by their argumentation. For example, an evidence-based approach which uses vulnerabilities as arguments [18], or an approach which uses the source of security requirements as arguments [19].

2.4 Automotive Development Process

An Automotive Development Process (ADP) is very complex. It includes not only security aspects but also aspects that relate to functionality, cost, quality attributes and non-functional requirements. Figure 2.2 shows stages of an ADP designed by Hirz et al from 2013 [20]. Although it may no longer be state-of-the-art, it gives good insight about what goes into an ADP.

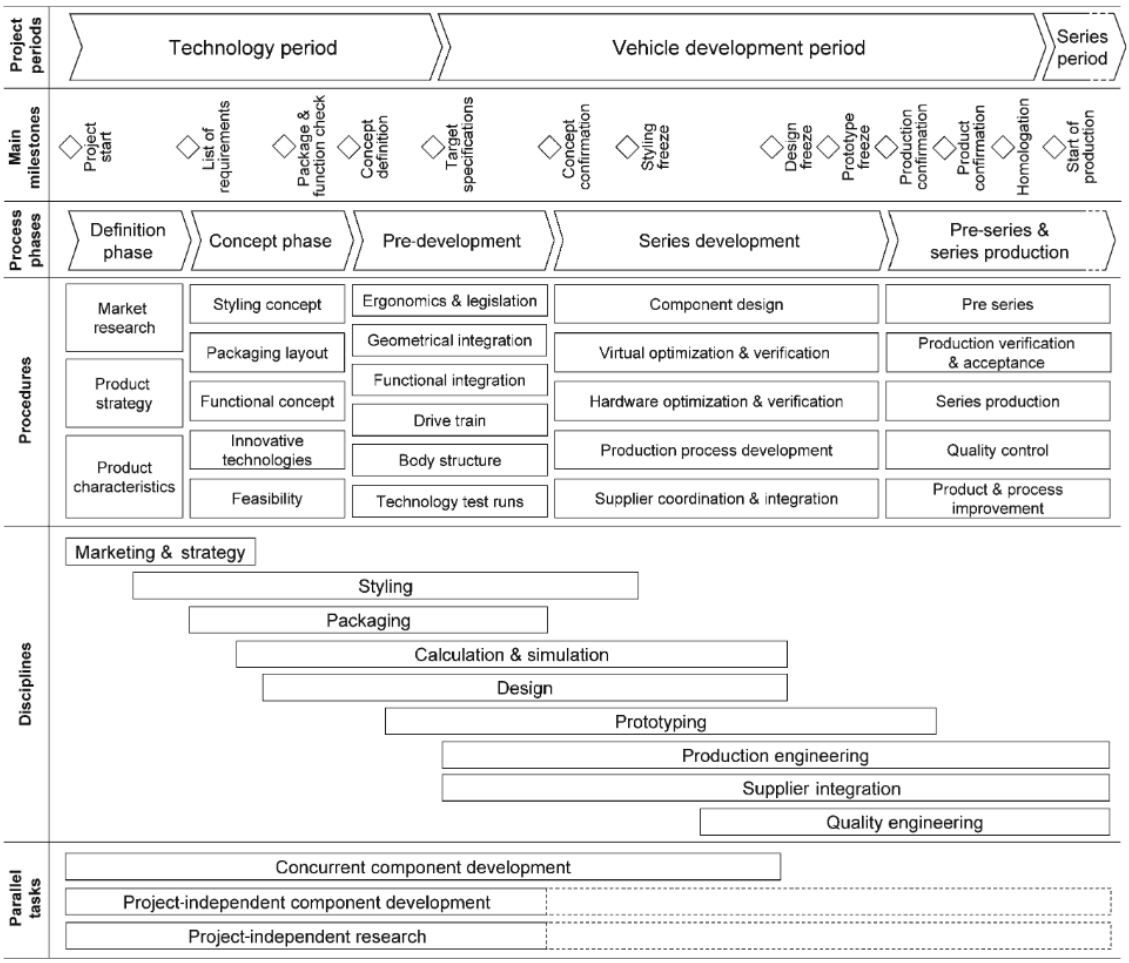


Figure 2.2: Stages of a typical state-of-the-art full-vehicle development process from 2013 by Hirz et al.

From the Definition phase, where activities such as market research and product strategies are defined, up to Pre-series & series production phase, where activities such as quality control and product verification & acceptance are performed. It manages to display the complexity of an ADP, despite not being up to date with current standards. Given that the figure is from 2013, we can only assume that security-related activities (like creating a SAC or conducting a TARA) would be part of the Series development and Pre-series & series production phase if the ADP was up to date. Perhaps even part of the Pre-development phase to conduct a TARA on a higher level of the system.

In this thesis, an artifact found in an ADP is something in an automotive system that can be used to create claims and arguments in a SAC, as well as evidence to support the claims.

2.5 Further technical background

In this section we further elaborate on terms and concepts that were used in this thesis.

2.5.1 Electronic Control Unit

In the automotive industry today, a lot of new features are being integrated into vehicles, the most popular being the self-driving feature. Most of these features are integrated as an electronic system or a system of systems. To coordinate these electronic systems, vehicle manufacturers and Original Equipment Manufacturers (OEMs) have introduced Electronic Control Units (ECUs) [21]. Today's internal architecture of the vehicles is quite complex and can be distributed over more than a hundred ECUs [22]. Currently, there is a transition towards a more centralized architecture where the functions will be concentrated on much fewer and more powerful ECUs [23]. These central ECUs are connected to sensors, actuators, external communication media, and to some smaller legacy subsystems [22] where they are used to perform different actions. This is why they are considered an asset, among other assets. An ECU takes inputs from sensors and compute the data for its required task. It can also take input from another ECU to perform its tasks.

2.5.2 Threat Analysis & Risk Assessment

Threat Analysis & Risk Assessment (TARA) is a risk assessment method/framework and a key activity defined by ISO/SAE-21434. Conducting a TARA consists of three blocks: performing a risk analysis, selecting risk treatments, and deriving cybersecurity goals [6]. The basic steps to performing a risk analysis and selecting risk treatments are:

1. Identify assets and their damage scenarios
2. Analyze the identified threats
3. Analyze risks by looking at the damage potential and the likelihood of each threat
4. Reduce risks with mitigation strategies

Based on that analysis, you can derive the cybersecurity goals for the assets.

2.5.3 Confidentiality, Integrity, and Availability

Confidentiality, Integrity, and Availability (CIA) are three security principles considered to be foundational to any security program. Any time there has been a cyberattack (hacking of account, data leak, etc.) it is most likely due to one of these

principles being violated [24].

Confidentiality refers to the effort to keep data private and secret by, for example, controlling access to data. This usually entails protecting sensitive information by implementing access levels for said information. The sensitivity of the data can be measured by how much damage it would cause if it were breached. Integrity refers to the quality of the data. In particular, protecting the data from being deleted or manipulated by an unauthorized party [25]. "It is correct, authentic, and reliable" [24]. Lastly, Availability refers to the extent to which the data that is being protected can be accessed. This means that authorized users are able to access systems, applications, and data when they need to [24].

3
Related Work

In this section we mention work that are related to our thesis. Different papers are referenced that have done similar work to ours to some extent, however there are some differences that separate our work from others.

In the work done by Finnegan et al. [10], the authors introduce a framework for creating a security case and also conducted a mapping between identified security controls from security related standards and the security capabilities. This was done in the Medical Device (MD) domain and did not use any of the standards that we have used in this thesis.

Similarly, the work by Ali et al. [26] is connected to the software engineering domain. Their work was focused around Zen Cart, a web-based e-commerce open source software, where they derived a threat model that generated a list of threats that are all relevant for the software. They also developed a plug-in for Eclipse that lets them create an SACs but do not mention if they follow a particular approach. Our thesis presents two SACs that were created using the CASCADE approach without the use of any software or plug-ins, except for diagrams.net [27], a diagramming application, to visually represent the cases. Additionally, the threats listed in our catalogue have been analyzed and mapped to specific assets that are relevant to the automotive industry.

The systematic literature review (SLR) by Mohamad et al. [4] shares the same goal as our thesis, which is to support the creation of SACs. The authors review over 50 papers in this topic and provide a body of knowledge that gives insight into SACs and its creation, including providing a workflow for SAC creation. While the research goal was the same, the aim of how to achieve the goal was different. The SLR resulted in informative artifacts, whereas our work yielded some informative artifacts but mostly applicative.

Apart from the papers mentioned above, there is a lot of research in the topic of Security Assurance Cases that contribute to the topic in their own way, such as the SLR by Mohamad et al. and the papers they reviewed. In our research, we do that by building open source catalogues, conducting a gap analysis, and creating a Security Assurance Case report template.

4
Methodology

In this section, the Design Science Research (DSR) methodology is introduced with a brief background and how the method is applied. We then explain how the results are achieved following the DSR method.

The structure and procedure of the thesis study was split into two iterations. Each iteration used the DSR methodology by Hevner et al., using the model by Vijay Vaishnavi and Bill Keuchler [28]. Figure 4.1 shows a brief look of DSR process model and the steps involved [28].
The process steps, as shown in the figure, are as follows:

1. **Awareness of the problem**
   The awareness of a problem is related to identification of problems within a reference or perhaps coming from multiple sources. The outcome of this step is a proposal (formal or informal) for a new research effort.
2. **Suggestion**
   The suggestion phase includes the creation of a tentative design of the proposal, which is based on either existing, or new and existing elements. The outcome is a prototype that leads into the development step.
3. **Development**
   The tentative design is further developed and implemented in this step. The outcome here is the artifact that is developed which will be further evaluated in the next step.
4. **Evaluation**
   The developed artifact is evaluated here by considering qualitative and quantitative research methods. The evaluation is used for improving the results and refining the design of artifacts based on the evaluation of the artifact.
5. **Conclusion**
   The conclusion step is the end of the iteration cycle or is the finale of a specific research effort, where the results are presented.

4.1 First Iteration

In the first iteration, the focus was on identifying artifacts in the automotive development processes that can be used for creating SACs. The research, preparation, and development of the artifacts from this iteration is explained in detail in this section.

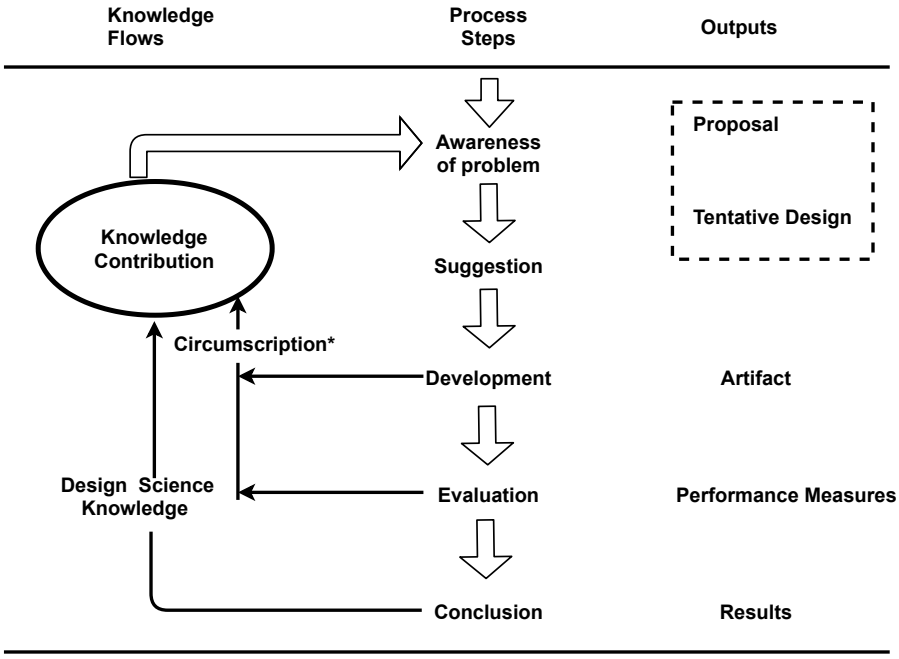


Figure 4.1: Design Science Research Process Model

4.1.1 Awareness of the problem

To bring awareness to the problem, a literature review was conducted on SACs, ADPs, and the situation of these two in the real-world. The systematic literature review by Mohamad et al. [17] was used to gain an initial understanding of SACs and their applicability in the real-world. The literature review includes over 50 papers about SACs. Throughout the paper, snowballing was performed on different levels to gain a deeper understanding of the problem and topic. The papers Asset-driven Security Assurance Cases with Built-in Quality Assurance [29] and REMIND: A framework for the resilient design of automotive systems [22] were found which held information about assets in a ADP.

The UNECE WP.29 GRVA [7] standard was studied, in which a list of threats and vulnerabilities, as well as their mitigations. Furthermore, the standard had referenced to both ISO-26262 and ISO/SAE-21434 which further confirmed the validity of the standard. In terms of finding evidence, the literature review could not provide the information we were looking for. We set out to find concrete evidence for systems and assets, but realized that it would not be possible to generalize such findings for past, current, and future claims. Instead, we focused our attention towards finding types of evidence. A technical report by Lipson et al. [30] was found that mentioned three categories of evidence and the different types under each category.

We realized that the information we require for our catalogues is available, however, it is from different bodies of knowledge making it difficult to find this information readily available.

4.1.2 Suggestion

The paper on Asset-driven SACs and REMIND lists categories of the types of assets that exist in a ADP. A short comparative study was done to compare the difference in the assets mentioned in those papers. We found that although the aim of the papers was different, the assets listed were very similar. Additionally, an investigation was done on the UNECE WP.29 GRVA standard in order to identify whether the vulnerabilities could be mapped to the identified assets in the papers. This resulted in the conclusion that an analysis would be needed to map the assets to the vulnerabilities and controls.

As identified in the previous phase, the report by Lipson et al. mentioned three categories of evidence. These were Actor, Process, and Technology. Each category had a number of attributes assigned to them. For example, the Actor category had attributes Competence, Capacity, Trustworthiness, Objectivity, and Resources. These attributes (types of evidence) were studied in order to better understand them and gather them into a catalogue.

In order to create the templates for catalogues needed for this project, the service Google Sheets [31] was used. The Sheets were prepared before-hand in order for the data gathering to be more about the extraction rather than how or where to place them.

4.1.3 Development

Since the research is being conducted by two people, two separate 'analyses' were conducted. These analyses were then brought together and discussed to bring forth the final version of the catalog, one where both of us had agreed upon the vulnerabilities of the assets listed. This was helpful to remove a lot of the bias that would come from only one person conducting the analysis.

In terms of evidence, the categories and their attributes found in the paper by Lipson et al. [30] were used to create a separate catalogue for evidence categories and types. In the paper, these categories had attributes that had been assigned to them which one could use to help determine the correct type of evidence for a claim.

One strong point about these catalogues is that they are available, collected and structured, outside of the sources they come from, so that anyone interested can use them almost immediately.

4.1.4 Evaluation

During the presentation of the catalogues to the case company, it was shown that a large part of the catalogue made in the development phase had already been implemented into their tool, SystemWeaver, which was unfortunate due to the work/effort that had already been done by both parties. However, contrary to the catalogue in SystemWeaver, we had references to mitigation strategies included in our cata-

logue. The authors went on to implement the remaining parts of their catalogue into SystemWeaver. In regards to the evidence catalogue, SystemWeaver did not have such a catalogue. Therefore, it was very interesting for them to include it in their tool. The authors successfully implemented that catalogue into SystemWeaver as a GSN graph, which ended up having an unexpected and interesting use in the system engineering tool.

4.1.5 Conclusion

In this iteration, to bring awareness to the problem, the literature review showed great results as it helped identify key elements for the creation of SACs to a certain extent. Evaluations done in focus groups were fruitful as they provided insight into what is starting to emerge in the industry and what is still missing. In conclusion, the catalogues made and implemented have proven to be of great interest to the case company and were useful in the second iteration of the research.

4.2 Second Iteration - RQ2

In the second iteration, the focus was on gap analysis by identifying the gaps between literature and industry. Two aspects were taken into consideration: (1) What are the suggested approaches in literature when creating SACs and, (2) How well do the ADP cover the needs of the approaches suggested in literature. How this was researched is further explained in the sub-sections below.

4.2.1 Awareness of the problem

During the literature review in the first iteration, subject matter experts at the case company were asked about the creation of SACs in practice. It was shown that there is no proper process of creating SACs and that there is a lack of documentation right now. Although a lot of research has been done on approaches to creating SACs, as well as SaACs (Safety Assurance Cases), SACs and their creation have not been adopted in the industry. For that purpose, we studied some of the existing approaches in literature to find one suitable for our purpose with the examples given to us in SystemWeaver.

Furthermore, the case company asked if a prototype for a SAC report template could be made so that they could use it when presenting SACs to their customers. For that purpose, we also studied past templates the case company had made to get a better understanding of what they were looking for.

4.2.2 Suggestion

The SLR was found to be useful in this iteration as well, having identified 26 different approaches from literature. There were many different approaches to choose from with varying drivers, i.e., the main elements that the approach uses to create a SAC. A study was done in order to find approaches that fit the systems we had

to create the SACs for, as well as provide some knowledge about the needs of the approach compared to what was available in the ADP.

The case company had some existing examples that were used to create SACs. Two examples were used in order to document the creation of the SAC while also taking notes about gaps that were found. One example was the Headlamp item example from the ISO/SAE-21434 standard [6] and the other example was a Cooperative Adaptive Cruise Control (C-ACC) system. The examples were in the form of TARA grids which had listed assets, violated properties, threats, and more.

The SAC report template did not start until after we had created some SACs ourselves. This was to give us a better understanding of the structure of the SAC, what's important to show and how it could be shown. A short workshop was held by the case company where we learned about the XML scripting language that is used to create the documents/reports in SystemWeaver.

4.2.3 Development

From the earlier iteration, we had found the paper about Asset-driven SACs [29]. The study of this paper is actually about an approach called CASCADE. As mentioned in the title, it is asset-driven and includes a structured way to build security cases with the assets you have. We decided to use this approach to create the SACs, as it fits well with how the examples were provided in SystemWeaver. This approach is further explained in section 5.2.

The first example used to create an SAC was the Headlamp example. It was an easier example to start with since the Asset-driven SACs paper used the same example and the ISO/SAE-21434 had further information about the system. The SAC was created initially on a whiteboard, where each section of the approach was done independently and later put together in a diagram creation tool called diagrams.net (formerly known as draw.io) [27]. Although it was the easier example, the paper on Asset-driven SACs had summarized their example of the Headlamp item, leaving us to decide for ourselves how to move forward. Still, we made an attempt but it was not extensive.

The second example was done almost completely in diagrams.net. The example was larger and more detailed, and therefore more complex when following the chosen approach. This was mainly due to the fact that in SystemWeaver, they had split the system into two levels of TARA. A Level 1 TARA had a higher level of abstraction resulting in security goals, and a level 2 TARA which was done on a lower level which resulted in security requirements. These two levels were linked by the security goals (the outcome of level 1 and one of the drivers for level 2).

Applying CASCADE on the examples provided by the case company was not only done to create SACs but also to evaluate how much the approach covers the needs

of the ADP, and to assess the supporting quality of the artifacts from RQ1. Occasionally, we would consult our catalogues to look-up the assets being used in the examples and try to compare the vulnerabilities and mitigations the catalogue had versus the ones included in the example. One reason was to see if we could, in fact, identify any missing assets, vulnerabilities or controls. However we also wanted to see whether the catalogues were truly useful.

The development of the SAC report template took some time but, with the cooperation of the case company, a successful template prototype was made. The prototype was made in the system engineering tool using the XML scripting language. The code was written so that it would work for any SAC they would create in their tool.

4.2.4 Evaluation

The security assurance cases were presented to the case company for evaluating the results. The questions that were asked in the focus group related to the SACs applicability in an industrial setting, the understandability and accuracy of the approach, how the created SACs could be augmented to further facilitate the integration to the organisation, and the current support that the company has for creating the necessary documentation to utilise the approach.

The report template was also assessed. It lacked some finesse due to the lack of experience in terms of designing a report template using XML, but the overall outcome was seen as positive to both parties. There were some concerns about the approach which will be addressed in later sections.

Finally, we got in contact with the person from the automotive industry to confirm whether the gap is true for them and the company they may work for. We could not get in a meeting with them but we managed to send a questionnaire that they replied to. The summary of questionnaire will be addressed in section 5.2.2.

4.2.5 Conclusion

We applied the CASCADE approach to existing examples, which yielded interesting findings. We studied previous report templates produced within SystemWeaver to go on and create the first SAC report template at the case company. The evaluation sessions resulted in some proposed changes to the SACs which further revealed a gap between research and industry. The changes were made and implemented by an expert at the case company and the feedback was noted.

5
Results

5.1 RQ1 - Catalogues

In this section, we report the findings of our study based upon the methodology we applied to gather the information.

5.1.1 Assets, Vulnerabilities, and Controls

The first catalogue includes assets that are mapped to their corresponding category. Each asset is then mapped to a vulnerability reference which is in turn mapped to a control (mitigation strategy) reference. The initial idea was to only use this catalogue while building a SAC, but it was found useful post creation as well. During the creation of a SAC, you could use this catalogue to identify missed vulnerabilities and/or controls. One could find the type of asset one is working with and trace it with the listed vulnerabilities and control references in the catalogue. Combining the catalogue with the UNECE WP.29 GRVA, where the reference numbers for the vulnerabilities and controls can be mapped to their descriptions, results in an artifact that supports the creation of SACs. Post creation, you can use this catalogue as a quality assurance by, for example, comparing the vulnerabilities and/or controls listed in your SAC with those listed in the catalogue to validate your claims, strategies, etc. You can find this catalogue, with both the first and revised version, online as a Google Sheet document.

Category	Assets		Vulnerability	Control
Network/Communication	**Internal Network/Communication**	CAN	2, 4, 5, 8, 10, 11, 15, 20, 24, 29	M3, M10, M11, M6, M13, M14, M23, M15, M18, M19, M7
		LIN		
		MOST		
		Automotive Ethernet		
	External Network/Communication	Wi-Fi (external)	1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 18, 20, 29	M1, M2, M8, M3, M10, M11, M6, M13, M9, M14, M23, M15, M17, M18, M19, M20, M22, M7
		Mobile Network (external)		
		V2X (external)		
		Bluetooth (external)		
		OBD-II (external)		
Data Storage	User Data		3, 5, 7, 9, 19, 20, 30, 31	M1, M2, M4, M5, M8, M6, M7, M10, M12, M24, M9
	Logs/Reports/Events		3, 5, 7, 9, 19, 20, 21, 22, 30, 31	M1, M2, M4, M5, M8, M6, M7, M10, M12, M24, M9
	Forensics data		3, 5, 7, 9, 20, 30, 31	M1, M2, M4, M5, M8, M6, M7, M10, M8, M12, M24, M9
	Cryptographic material		3, 5, 7, 9, 12, 19, 26, 30, 31	M1, M2, M4, M5, M8, M6, M7, M10, M8, M12, M24, M9, M11, M16, M23

Figure 5.1: Asset, Vulnerability, and Control Catalogue - Network/Communications and Data Storage categories

Category	Assets	Vulnerability	Control
Hardware	ECU (Hardware)	1, 2, 11, 14, 20, 24, 25, 27, 28, 32	M1, M2, M8, M3, M10, M15, M23, M7, M17, M13
	Sensors	2, 5, 8, 11, 14, 16, 20, 24, 25, 32	M3, M6, M10, M13, M15, M7, M17, M13, M20
	Actuators	2, 5, 8, 11, 14, 16, 24, 32	M3, M6, M10, M13, M15, M17, M13, M20
Software	ECU (Software)	1, 2, 5, 6, 9, 11, 12, 13, 14, 15, 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28	M1, M2, M8, M3, M6, M10, M9, M15, M11, M16, M23, M7, M17, M18, M19, M13, M20
	External Libraries	1, 2, 5, 8, 9, 11, 12, 15, 17, 22, 23, 28	M1, M2, M8, M3, M6, M10, M13, M9, M15, M11, M16, M23, M18, M19, M21, M7
	OS	1, 2, 5, 9, 12, 13, 14, 15, 23, 28	M1, M2, M8, M3, M6, M10, M9, M23, M11, M16, M17, M18, M19, M7
	Virtualization (Docker containers for example)	1, 2, 5, 8, 9, 12, 15, 17, 23, 28	M1, M2, M8, M3, M6, M10, M13, M9, M23, M11, M16, M18, M19, M21, M7
	Applications	1, 2, 4, 5, 9, 12, 15, 17, 22, 23, 28, 31	M1, M2, M8, M3, M10, M11, M6, M9, M23, M16, M18, M19, M21, M7

Figure 5.2: Asset, Vulnerability, and Control Catalogue - Hardware and Software categories

The first version of the catalogue is shown in Figures 5.2 and 5.1. The figures show a sheet with the categories Hardware, Software, Network/Communication and Data Storage. The assets in the categories are mapped with relevant vulnerabilities and controls. In addition to the columns that can be seen in those figures, the catalogue also has columns for: type of threat/attack the category would have, CIA properties, STRIDE properties, and vulnerability count for each asset as shown in Figure 5.3. STRIDE is a model of threats and it has relevant desired properties with respect to CIA. The vulnerability count is especially interesting when it comes to analyzing the results.

Type of Threat/Attack	CIA	STRIDE	Vulnerabilities
Disruption or **direct interventions** that influence their availability and integrity. For example: **fault injection**, **information leakage**. **Tampering** with existing hardware or **installing malicious hardware** into the vehicle can act as mediators to gain complete vehicle control. Input signals from sensors may be **manipulated to cause an unwanted behavior**.	**Hardware Availability** (disrupt or disabled components or system resources) **Integrity** (information leakage, extract secret keys).	- Hardware Denial of Service - Tampering	10 10 8

Figure 5.3: Threats and Desired properties for the Hardware category.

A revised version was made (Figure 5.4) after we received some feedback from the first version of the catalogue. It is not clear what the vulnerabilities and controls mean with only the reference numbers, as seen in the first version. The revised

version shows what the vulnerabilities refer to as well as a short description of the controls. We also added the affected assets so it would be as complete as possible.

Vulnerability Reference		Mitigation Reference		Assets
5	*Unauthorized manipulation via comm. channels*	M10	Verify authenticity and integrity of messages received	Sensors Actuators ECU (Software) External Libraries OS Virtualization Applications Internal Network/Communication External Network/Communication User Data Logs/Reports/Events Forensics Data Cryptographic Material
		M6	Miminize impact of an attack on vehicle ecosystem	
		M7	Protect system data/code via access control techniques	
6	*Unreliable / Untrusted communication source*	M10	Verify authenticity and integrity of messages received	ECU (Software) External Network/Communication
7	*Disclosed information*	M12	Protect confidential data transmitted and received	External Network/Communication User Data Logs/Reports/Events Forensics Data Crypographic Material
		M8	Disable access to personal or system critical data by unauthorized access	
8	*DOS via communication channels*	M13	DOS detection and recovery measures	Sensors Actuators External Libraries Virtualization Internal Network/Communication External Network/Communication
9	*Unauthorized access to vehicle systems*	M9	Unauthorized access detection and prevention measures	ECU (Software) External Libraries OS Virtualization Applications External Network/Communication User Data Logs/Reports/Events Forensics Data Cryptographic Material

Figure 5.4: AVC Catalogue revised.

5.1.2 Types of Evidence

A lot of research was done to find evidence that could be used in SACs, however we realized that it would be quite difficult to find evidence for claims that don't yet exist. The evidence that is required for the claims will most likely be unique to that claim. Add to the fact that this is still a topic that is relatively new, this kind of finding could be a study in itself. Instead, a catalogue was created based on a report about *types* of evidence in assurance cases, specifically in cyber security. The report, which this catalogue is derived from, is called **Evidence of Assurance: Laying the Foundation for a Credible Security Case** [30]. This report goes in-depth into the types of evidence, how to understand and gather them. The authors studied the report and extracted the categories and types, in order to quickly access that body of knowledge that would be useful when trying to provide evidence for SACs. Figure 5.5 shows the Actor and Process category in the catalogue, while Figure 5.6 shows the categories Technology and Product. This catalogue is also available online as a Google Sheet document.

Category	Attribute	Description
Actor	Competence	An actor's skills/expertise for a given task, or more generally, in a specific domain (e.g., credentials are one source of evidence)
	Capacity	How much can be accomplished within a given period of time
	Trustworthiness	This relates solely to intent: the actor's veracity, honesty, and alignment with the mission of the system
	Objectivity	Absense of conflicts of interest in a given context
	Resources	Assets (inlcuding economic assets of organisation)
Process	Capability	What can actors accomplish by using the process?
	Quality	How good is the process at achieving a desired result, be it analysis of code for a desired security property or the ability to construct a component with a high assurance of security? Quality arguments include conformance to best practices as embodied in recognized standards (or the more general claim of "adherence to industry standard practice" or "due care") and the existence of studies that validate the effectiveness of the process, as well as typically weaker arguments about the performance history associated with systems for which this process was used
	Cost/Benefit	How practical is the process, i.e., how much does the process cost with respect to the value obtained?
	Context of Use	What is the context in which the process is applicable and achieves valid results?
	Repeatability	Within its context of use, is the process readily repeatable over time across different project teams, across different organisations, even across different application domains (i.e., industry segments)? For example, is the process well documented and easy to follow? What organizational and individual resources are needed (e.g., skill sets and tools) so that the process produces the same results each time it is executed, regardless of who executes it?
	Reviewable	Does the process document intermediate steps (intermediate inputs and outputs), along with associated rationale, so that the entire process is reviewable (for example, by an independent third party)?
	Schedule, Workflow, and Resources	What actor resources (with what skill sets) and what technology resources are required to carry out the process, and for how long are they needed? What interactions are there among the individuals and workgroups (internal and external) participating in the process? How predictable is the schedule for the process? This attribute is closely related to the cost/benefit attribute above
	Accountability/Attribution	Does the process keep track of the actions of the participating actors, and are the appropriate actors accountable for their actions (i.e., are the results of significant activities attributable to specific actors?) For example, are test results dated, linked to the correct version number of the code, digitally signed, and stored in a database for future use as evidence of assurance?

Figure 5.5: Actor and Process Categories in the Types of Evidence catalogue.

The catalogue is first divided into the different categories. For each category, the types (the report calls them attributes) have been listed together with the description of that type. As seen in the catalogue, it is easy to locate the type of evidence you may want to consider for your claims. However, the types of evidence here is for cyber security as a whole and not specific to the automotive domain. Therefore, the supporting aspect of this catalogue comes under verifying or validating your evidence as they can be seen as recommendations or helpful evidence types to look out for when creating an SAC. Perhaps some of the attributes listed are exactly what the SAC needs, or maybe it is an attribute that was not considered.

It is worth mentioning here that the Product category was added post creation of the catalogue by the case company. The way they viewed this catalogue was similar to a requirement. Each of these attributes are treated as required evidence for the case in question and what is then missing is the SAC itself. Therefore, the Product category was added with the attribute *Validity & Completeness of Analysis* which

would provide the SAC. At this point it is understandable if it does not make any sense. This will be elaborated on further in the results of RQ2.

Category	Attribute	Description
Technology	Capability	What can the technology accomplish?
	Validity & Completeness of Analysis	*Has the anaylysis been done thoroughly? Do we have evidence for all the product argumentation?*
	Quality	Which software or system quality attributes (security, reliability, fault tolerance, etc.) are associated with the technology?
	Visibility	Are the artifacts of life cycle processes used to create the technology available to users, customers, certifiers and others, so that the technology's quality attributes can be assessed through analysis of those artifacts?
	Cost/Benefit	How affordable is the technology relative to its value?
	Context of Use	What is the context in which the technology is applicable and achieves valid results?
	Resources	What actor resources (with what skill sets), processes, and other technology resources are required to make effective use of the technology?
	Traceability/Accountability/Attribution	Does the technology keep track of the actions of the participating actors, and are the results of significant activities attributable to specific actors? Are significant events logged and available for audit?
Product	*Validity & Completeness of Analysis*	*Has the anaylysis been done thoroughly? Do we have evidence for all the product argumentation?*

Figure 5.6: Technology category in the Types of Evidence catalogue.

A study was conducted on whether all of the types of evidence were needed in the Automotive Development Process. All of the types listed can be useful for an SAC, but perhaps some of them stand out more than others. The study yielded the types shown in Table 5.1.

Categories	**Attributes/Types**
Actor	Competence
	Trustworthiness
Process	Capability
	Quality
	Repeatability
	Reviewable
	Accountability/Attribution
Technology	Capability
	Quality
	Context of Use
	Resources
	Traceability/Accountability/Attribution
Product	*Validity & Completeness of Analysis*

Table 5.1: Types of Evidence - Filtered

The process category yielded five types: Capability, Quality, Repeatability, Reviewable, Accountability/Attribution. It is important to make sure what actors can accomplish by using the process and how good is the process at achieving a desired result. The process will be repeatable within its context of use across different organizations, domains and project teams so that the process produces the same results when executed each time. The outcome produced by the process will be reviewable for example, an independent third party can review the entire process document. An example of a process category can be, a requirements document that provides the security concerns of the system.

The technology category provides five types: Capability, Quality, Context of Use, Resources and Traceability/Accountability/Attribution. The technology attributes are the same as process attributes. It is important to make sure that actors have the required skill sets, processes and other technology resources to make effective use of the technology. The quality of software is vital and is associated with technology. For example, quality attributes can be security of a component and reliability. An example of technology can be, Malicious control signals are detected, and prevented from being transmitted. This is related to technology and system since the technology is capable of performing detection and prevention systems of Malicious control signals.

The case company saw value in this catalogue and decided to implement it as a GSN graph in their system engineering tool. This kind of implementation, where the categories were top nodes and the types were child nodes, was really interesting. It was unclear for a moment how this would be used in the tool, but was made clear along the way and made sense when we were creating the SAC report template (section 5.2.1). Figure 5.7 shows how the catalogue is represented as a GSN graph in SystemWeaver. The technology evidence have five attributes as described in the figure. For example, G_102 shows context of use which means in what context the technology is applicable and achieves valid results. Other representations can be found in Appendix A (Figures A.1, A.2, A.3).

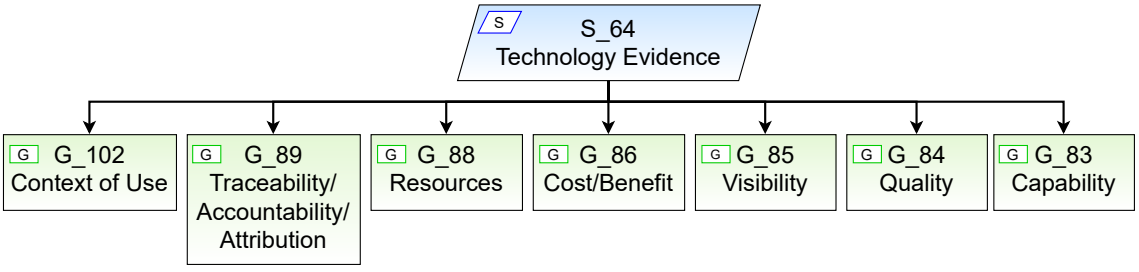


Figure 5.7: GSN graph representation of technology evidence.

5.2 RQ 2 - Gap Analysis

As mentioned in the suggestion part of this phase (section 4), the very first gap we identified was the fact that the industry had not adopted the creation of SACs.

Identified Gap 1:
The automotive industry has not yet adopted the creation of Security Assurance Cases.

This was identified from two sources: it was hinted in the SLR by Mohamad et al. [4], and the other from the case company. However, it was not necessarily true for other companies in the automotive industry. We got in contact with a person from the automotive industry that did not belong to the Systemite team. Thus, we formed a short questionnaire which was sent to them over digitally. The questions and a summary of the answers are available in the questionnaire section (5.2.2).

To further identify the gaps between literature and automotive industry, we used examples from the industry (SystemWeaver) and an approach from academia (CASCADE). Since the approach we chose uses the Goal Structuring Notation [13], which consists of different types of nodes and arrows, we started constructing the graphs on a whiteboard. The graphs produced were later made into a digital format that would make it easier to understand and read. Figure 5.8 shows the white-hat and black-hat block the C-ACC example SAC that we made in diagrams.net, a flowchart maker and online diagram software.

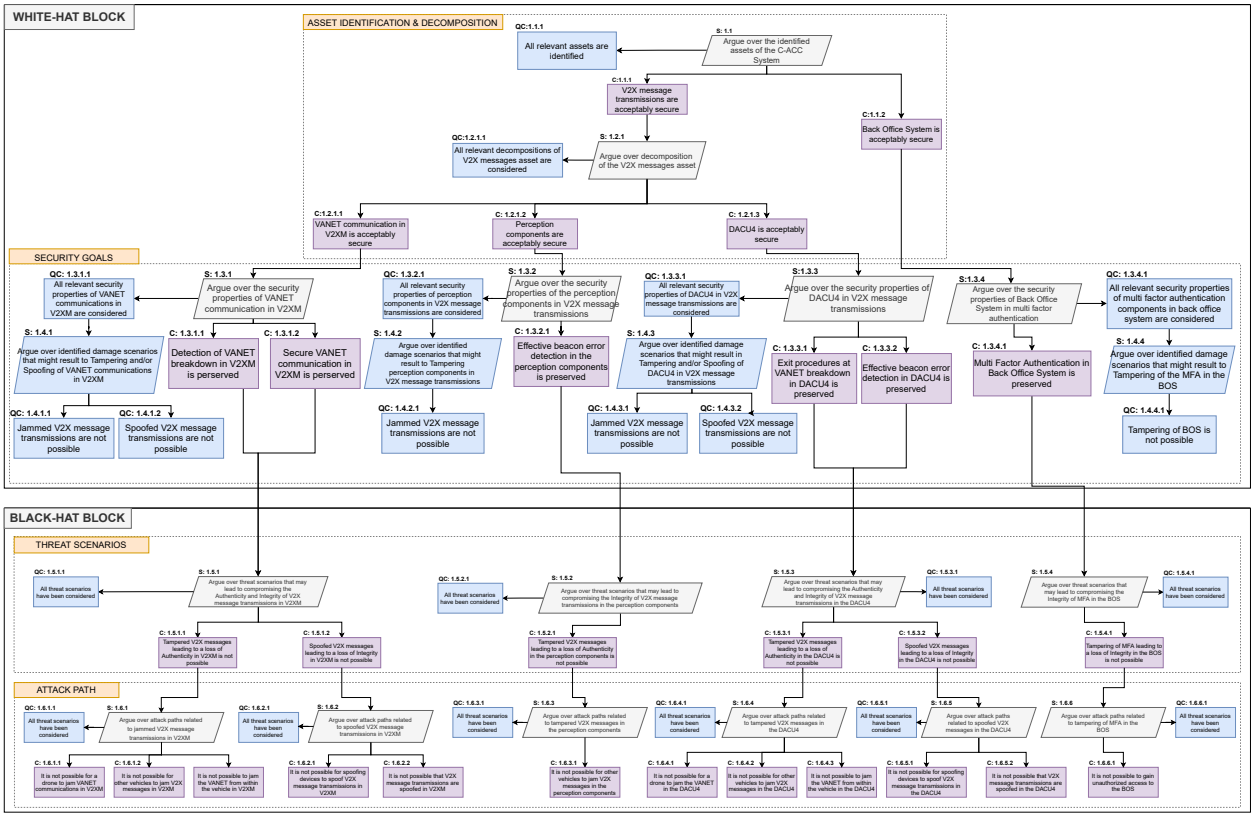


Figure 5.8: White-hat and Black-hat block from the C-ACC SAC showing the structure of the two blocks which are similar to the structure of CASCADE.

The figure shows an overview of the SAC for the C-ACC system that was designed in the same structure as the approach (see Figure 5.9). Since it is too large to show in this paper, Figure 5.10, 5.11, and 5.12 are cropped and zoomed in versions from the original to better show what the case arguments are. Other parts of the SAC are provided in the appendix in a similar manner. The first block, white-hat block, includes two levels: Asset identification & decomposition (AI&D), and security goals. In the AI&D part, you perform an analysis to identify the assets that are vulnerable to an attack. This is then further decomposed and linked to any identified sub-assets. For each identified asset, the claim is generally that the asset and sub-assets are secure. That is then linked to the second level of the white-hat block where we establish security goals for the claims from the previous level. We do this by identifying the relevant security properties (CIA) for the assets and create claims that represent the security goals.

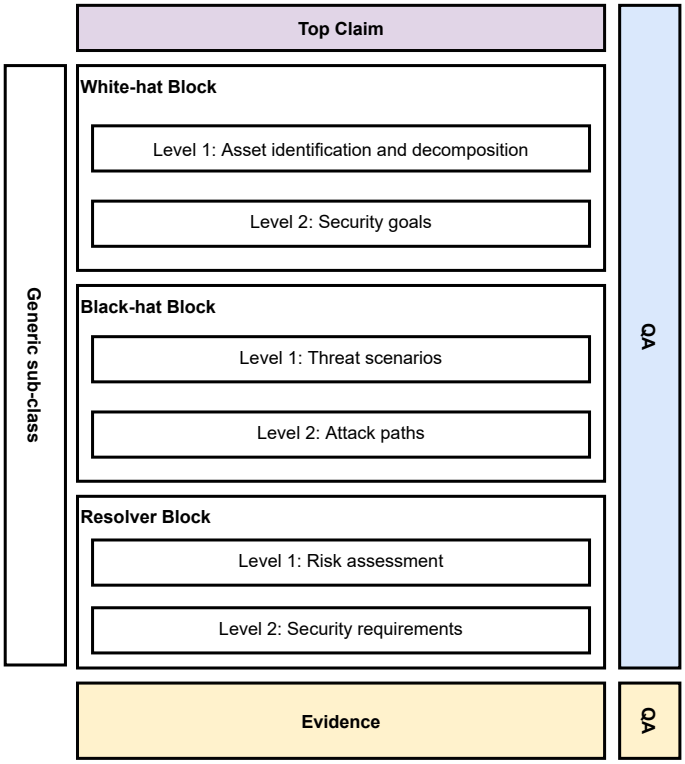


Figure 5.9: The structure of CASCADE approach.

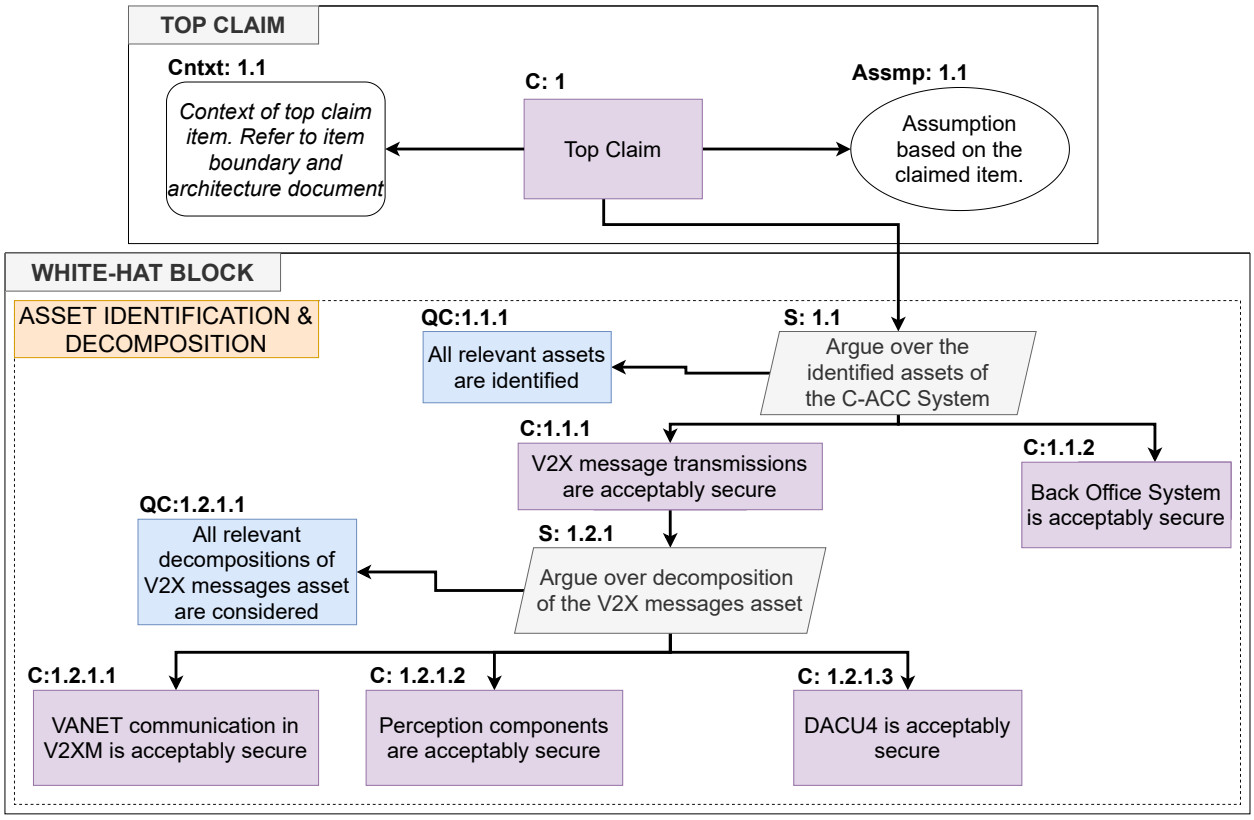


Figure 5.10: C-ACC SAC from Diagrams.net - Top item and part of the White-hat block. The path from the decomposed assets have been extracted into four paths, shown in Figure 5.11 and B.11, B.12, and B.12 in the appendix.

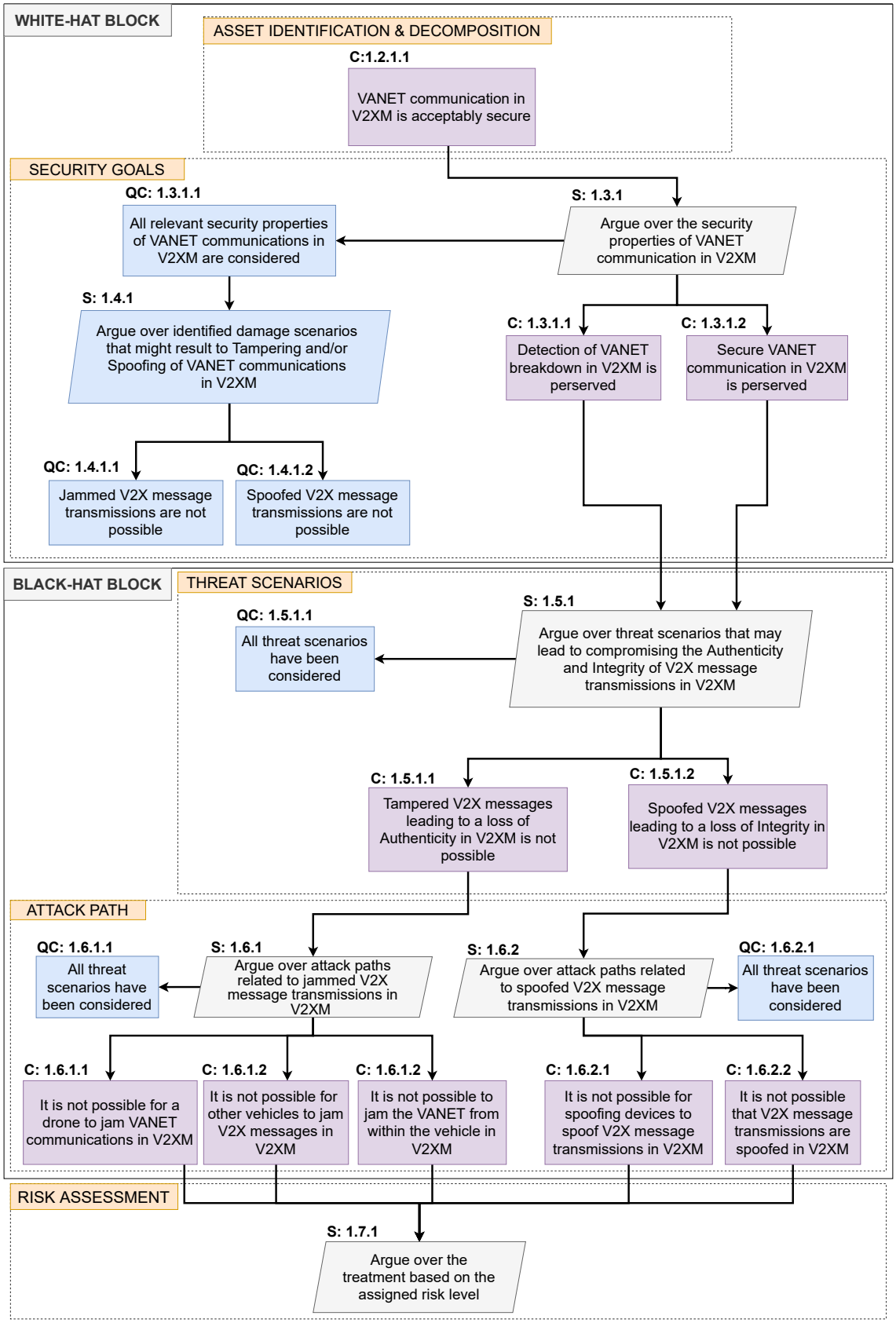


Figure 5.11: One of the paths from the V2X messages decomposition. Starts from the Security goal and goes all the way down risk assessment. For the resolver block, see Figure 5.12.

In our case, the main assets of the system were identified as Back Office System and Vehicle-2-everything (V2X) message transmissions (White-Hat Block, Asset Identification & Decomposition). The V2X asset was further decomposed into DACU4, Perception components and VANET communication due to their dependability. Since the Back Office System asset had no dependability with other assets and ECUs, it directly leads to level 2 of the white-hat block, security goals.

In the second level, the security goals were identified based on the example in SystemWeaver (White-Hat Block, Security Goals). Still, it is important to make sure that the relevant properties are covered when identifying security goals. In the real world, there would be an analysis of the security properties of the identified assets in level 1 that would lead to identified security goals.

In the black-hat block, the main goal is to identify threat scenarios that are linked to not fulfilling the security goals, as well as the attack paths that could lead to the threat scenarios. On the first level, threat scenarios, we create claims that are the opposite of the threat scenario happening, based on the security goals before. For example, if your security goal is "Integrity of a ECU X is preserved", then the claim of the threat scenario would be the opposite of that happening, i.e., "Spoofing of a signal leading to loss of integrity of ECU X is not possible".

For this, we consider damage scenarios that lead to compromising the security goals. For example, "Spoofed V2X message transmissions are not possible" which could possibly damage the authenticity of the asset.

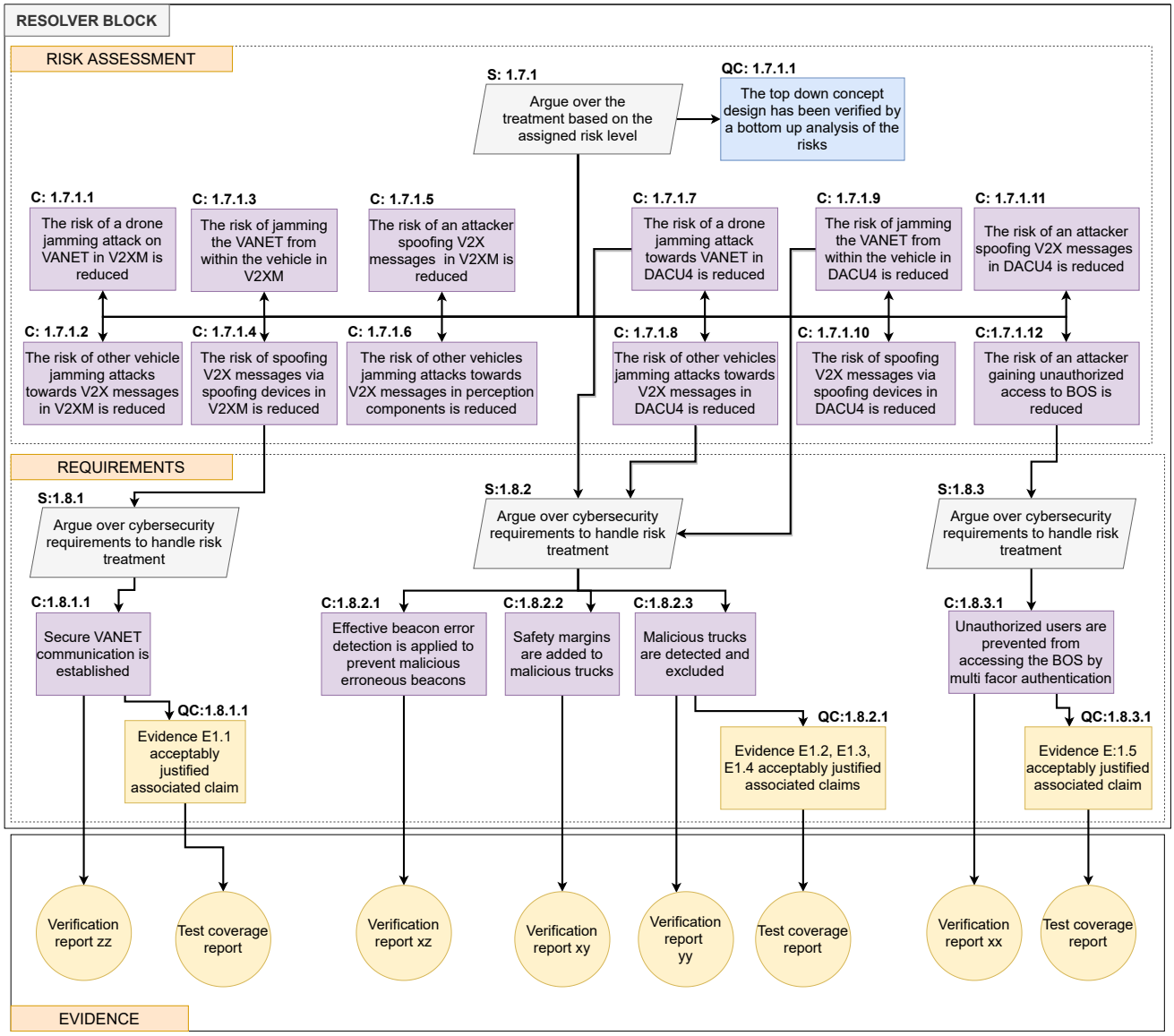


Figure 5.12: C-ACC SAC from Diagrams.net - Resolver and Evidence Block.

The main goal of the resolver block is to perform a risk assessment on the identified attack paths from the previous block (Figure 5.12, Resolver Block). Based on the risk level, we can choose how to treat the risk. Additionally, the requirements of risk treatments identified in the previous level are then expressed as claims. For example, if the attack path is "It is not possible for a drone to jam VANET communications in V2XM", then the claim of risk assessment would be to reduce the risk of the attack path happening, i.e., "The risk of a drone jamming attack on VANET in V2XM is reduced".

In the evidence block, the main goal is to assign evidence to claims. It is important to make sure that evidence is provided and the corresponding claims can be considered as justified. For example, a claim can be covered by any test report and verification report based on the type of the claim. We provide verification report and test coverage report for the claims that are derived from the attack paths to

the evidence. We use placeholders due to missing evidence in SystemWeaver.

A similar SAC was created for the Headlamp example which can be found in the Appendix 2.

During the creation of the C-ACC system SAC, we were placed in situations where we did not know how to proceed due to the complexity of the example. The example in the CASCADE paper was the only example we that, unfortunately, was not enough to aid us. Thus, some decisions had to be made on our part as we continued. Although it perhaps cannot be identified as a gap, but we think more examples of SACs would be extremely helpful for future practitioners in case they also end up in the same situation as us.

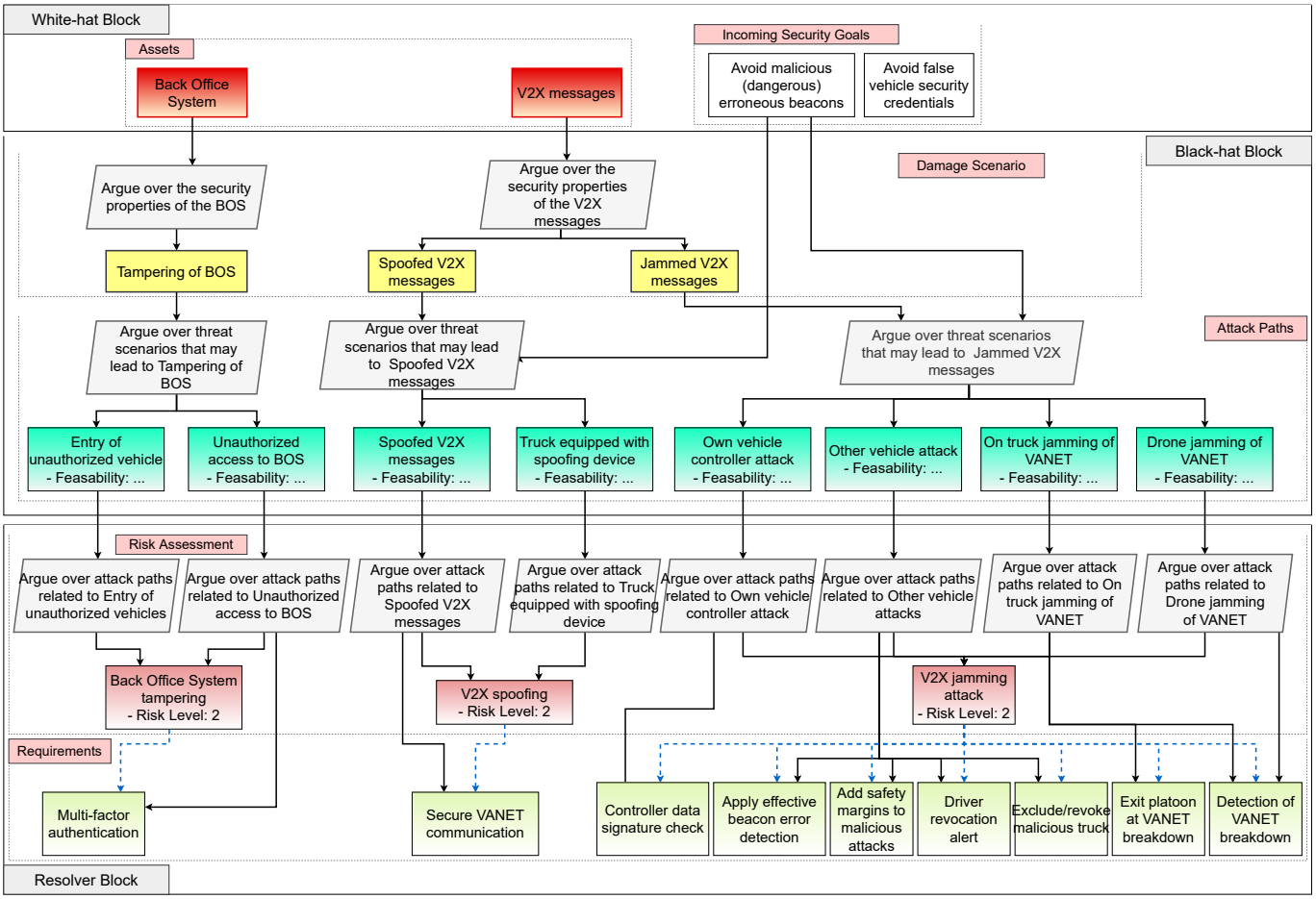


Figure 5.13: Redrawing of the C-ACC SAC from SystemWeaver.

Once we had evaluated the SAC with the case company, they implemented it into SystemWeaver. The same XML language was used to create the graphs represented in the system engineering tool, however, the resulting graph was quite different from the one we had made in diagrams. As seen in Figure 5.13, there are some differences between the one created by us and the one that is represented in SystemWeaver. The figure was redrawn for the purpose of readability, however a copy of the original can be found in Appendix B. Isolated parts of the figure can also be found in the

Appendix. During an evaluation meeting with the case company, it was discovered that incoming security goals were preferred as drivers, which is not suitable for the approach that we chose. The incoming security goals are actually a result of conducting a Threat Analysis & Risk Assessment (TARA), which is a risk-based activity. This raises the question:

Identified Gap 2:
How can the automotive industry use the outcome of risk-based activities to create security arguments?

Systemite conducts TARA analyses for the systems they have in SystemWeaver. This is an important part of the Automotive Development Process since you need to identify which assets are vulnerable, analyze the threats and risks, and finally create some security goals based on the analyses in order to mitigate the vulnerabilities. In order to capture all important information about a system, a TARA can be conducted on two levels. On a higher level of abstraction (level 1) the TARA is driven by abstract assets, resulting in incoming security goals. On a lower level (level 2), the TARA is driven by the incoming security goals from the previous TARA. The second level yields necessary requirements/security goals that trace all the way back to the level 1 TARA. Figure 5.14 shows a TARA grid of level 2 on the C-ACC system.

Asset	Violated Property	Threat	Damage Scenario	Safety Impact	Privacy Impact	Operational Impact	Financial Impact	Impact Level	Threat Argument	Attack	Feasability	Risk Level	Security Goal	Requirement	Affected Asset
V2X Messages	Tampering	Jammed V2X messages		Negligible	Negligible	Major	Negligible	Major		V2X jamming attack	Medium	3	Avoid malicious (dangerous) erroneous beacons	Apply effective beacon error detection	Drive Assist Unit, Perception components (Radar, Camera)
														Exclude/revoke malicious truck	Drive Assist Unit
														Add safety margins to malicious trucks	
														Driver revokation alert	
														Controller data signature check	
														Exit platoon at VANET breakdown	
														Detection of VANET breakdown	Vehicle to Vehicle Communication Manager
	Spoofing	Spoofed V2X messages		Moderate	Negligible	Moderate	Negligible	Moderate		V2X spoofing	Medium	2		Secure VANET communication	
Back Office System	Tampering	Tampering of Back Office System	• False C-ACC trucks entered in BOS • False C-ACC trucks allowed into C-ACC platoons • False C-ACC trucks enabled	Major	Negligible	Major	Major	Major		Back Office System tampering	Low	2		Multi factor authentication	Back Office System

Figure 5.14: C-ACC TARA Level 2

For each TARA, there is even an Attack Event grid which holds information about the type of attacks on the assets, the attack events, the risk levels, and more. Prior

to creating the SAC, we decided that we would simply use the second level TARA as our base for the SAC. However, notice that there is no information in the 'Damage scenario' column for this level. We had to consult the level 1 TARA a couple of times and map it with the second level, mainly for the black-hat block. Additionally, there is further important information for the black-hat block in the Attack Events grid. We started the SAC thinking that the security grid shown in the figure would be enough, however we ended up using the Attack Events grid as well as both grids from TARA level 1. In total, we were using 4 different sources of information for the SAC, although they were for the same system.

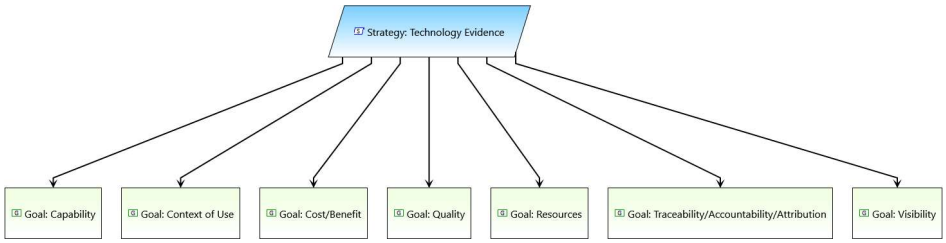
5.2.1 SAC report template

The creation of the SAC report template occurred after we had done the SACs for the examples. Once we had gotten the hang of the structure of the SAC and the approach, CASCADE, we could try to implement it into the template as well. As seen in the catalogue, we have categories for the evidence, as well as the types of evidence for each category. They were modelled as a GSN in SystemWeaver so that the report template could present that information in structured way. We have taken out page 5 from the document (Figure 5.15) to show how we designed it. There is a section for each category which first shows the GSN graph with the category shown as the top node, and the types shown as child nodes. We then use tables and columns to describe what each node is and what it represents, in the context of providing evidence for it. We would like to add that the examples provided for the types of evidence are just that, examples, and that they should not be the only documents/analysis/reports that are considered.

We mentioned earlier how, in the Types of Evidence catalogue, the Product category was added to hold the information about the SAC. Page 7 (Figure 5.16) shows how we have represented the SAC as a part of this report template. We start with a diagram of the full Security Assurance Case. Each section after that follows the same structure of CASCADE, starting from the White-hat block and ending at the Resolver block. We do not include an evidence block since the evidence are to be provided in the sections before, such as in page 5 from earlier. It is also important to note that this is just a first prototype of the template, thus it would need to be refined some more before we would have a version 1. Some diagrams may look small, especially the full SAC, but since the document is saved as a PDF it is easy to zoom in to read the details. This is intended and how Systemite wanted it to be implemented.

3.1.3 Technology Evidence

Claim diagram:



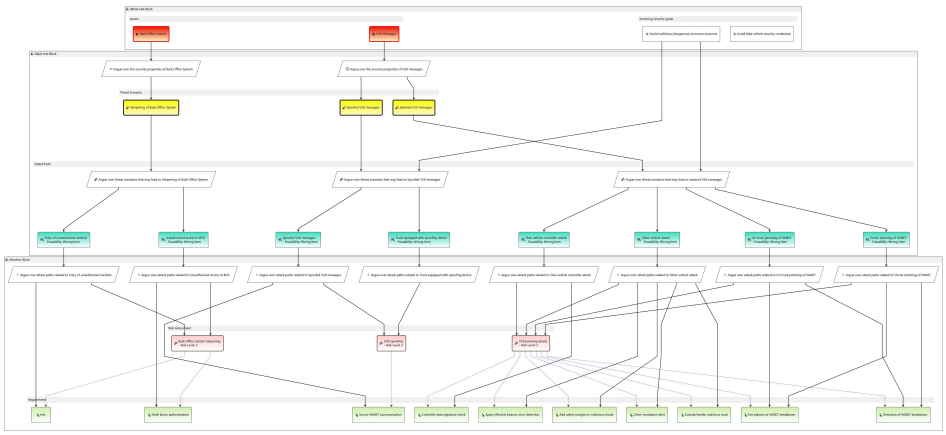
Strategy table:

Strategy Name	Strategy Description	Strategy Goals
Technology Evidence	The technology being used in the development process, for example, Electronic Control Units. Evidence in this category should include test reports, analysis, documentation, etc., that support that the technological assets are secure. Capability, quality, and traceability are some of the types of evidence to gather.	Capability
		Context of Use
		Cost/Benefit
		Quality
		Resources
		Traceability/Accountability/Attribution
		Visibility

Goal table:

Goal Name	Goal Description
Capability	What can the technology accomplish? Example(s): Test Results
Context of Use	What is the context in which the technology is applicable and achieves valid results? Example(s): • Use Case Report • System Architecture • Item Boundary
Cost/Benefit	How affordable is the technology relative to its value? Example(s): • Cost/Benefit Analysis
Quality	Which software or system quality attributes (security, reliability, fault tolerance, etc.) are associated with the technology? Example(s): • Quality Attribute Analysis/Report
Resources	What actor resources (with what skill sets), processes, and other technology resources are required to make effective use of the technology? Example(s): • Actor, Process • Technology Assets
Traceability/Accountability/Attribution	Does the technology keep track of the actions of the participating actors, and are the results of significant activities attributable to specific actors? Are significant events logged and available for audit? Example(s):

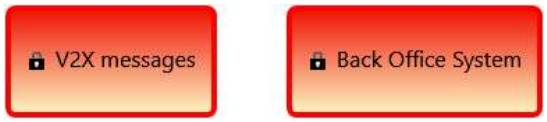
Figure 5.15: Page 5 from the SAC report template showing the Technology Evidence section.



White hat
This block is about the assets and security goals. Asset identification is done by conducting an analysis to find the artefacts of the system that are likely to be subject to an attack. The security goals come from identifying relevant security properties for the assets. Most often you would look into the Confidentiality, Integrity, and Availability (CIA) triad.

White Hat Diagram

Assets



Asset Name	Asset Description	Asset Type
Back Office System		Security System Component
V2X messages		Interface

Incoming Security Goals

Incoming Security Goal	ISC Description
Avoid false vehicle security credentials	Vehicle SCMS information in Back Office Systems shall always be correct. (Prevention, Detection)
Avoid malicious (dangerous)	Erroneous or missing data on dynamic state of other vehicles must not occur.

Figure 5.16: Page 7 from the SAC report template.

5.2.2 Questionnaire

As mentioned earlier section 4.2.4, we prepared a questionnaire and sent it to Örjan Askerdal, a Principal Engineer at AB Volvo who has previously done research on Security Assurance Cases. We got his consent to share the answers in this report.

Below you can find the summary of questions and answers.

Q1: Before your work in the field of Security Assurance Cases, to what extent did your company use or create SACs?
  A: They are not using SACs yet. However, they do use Functional Safety Cases which includes vehicle functionality such as steering, starting engine, headlamps, etc. So when it comes to cyber security, they have in mind a framework that will enable them to adopt SACs without any major problems.

Q2: Since then, has your company started/extended using or creating SACs as part of their development process?
  A: They have not, but they are looking into it since it is required by ISO-21434.

Q3: If your company is currently not using/creating SACs, have you looked at different methods/approaches of creating SACs? How far along is your company into adopting SACs as part of their development process?
  A: R-155, the UNECE legal requirements for cybersecurity, demands that OEMs have a Cyber Security Management System (CSMS) for newly developed parts in 2022, and for all parts being manufactured in 2024. The company aims to implement a CSMS by following the ISO/SAE-21434, that also demands Security Assurance Cases. Therefore, they plan to have SACs for all newly developed parts from 2022 and onwards. Currently, when it comes to claiming that products are cybersecure, they take a risk-based approach, i.e., they identify all the risks and mitigate the ones that are considered too high to accept as-is. However, an asset-driven approach is natural as it usually is where product development starts, thus it makes sense to structure an SAC around a certain asset.

Q4: If not 3, what is hindering your company from doing so?
  A: Automotive development is a very complex process not only dealing with cybersecurity but also dealing with functionality, cost, delivery times, and much more. This means all changes are complex, and they cannot be implemented without considering how they affect the rest of the development process.

With the answers provided, we got a little insight into Security Assurance Cases and the Automotive Development Process for that specific automotive company. Through this questionnaire with a well-known third party OEM, we gain validation (to a certain extent) for both of the identified gaps: **The automotive industry have yet to adopt the creation of Security Assurance Cases** and **The automotive industry conducts risk-based activities**.

6
Analysis & Discussion

6.1 Analysis

In this section, we further elaborate on how we performed analyses on our findings in both RQ1 and RQ2.

6.1.1 RQ1

The artifacts were gathered from different sources and mapped into a catalogue. We performed two separate analyses while mapping the assets to vulnerabilities and controls. During the evaluation of the separate catalogues, we realised that we had similar results with our mappings but there were still a few differences. To tackle this, a discussion session was held to combine both catalogues and produce a final version of the catalogue. This was done to remove bias and make sure that the results are reliable to some extent.

Having mapped assets to vulnerabilities, we counted how many vulnerabilities each asset had. Figure 6.1 displays the charts of vulnerabilities count against each asset. The charts shows all the assets from each category: Hardware, Software, Network/Communication, and Data Storage that have been colored blue, orange, red, and green, respectively, to show which category the asset belongs to. Additionally, we have the figures in Figure 6.2 where the larger chart has been separated for each category. Here you can look at number of vulnerabilities reported in each asset.

We found that the ECU (Software) asset had the most vulnerabilities, with a vulnerability count of 21 (Figure 6.1). The External Network/Communication asset was just behind with a count of 16. When it comes to cyber security, it makes sense to us that these have the highest counts, especially external network/communication assets, as we are moving towards an age where everything is connected via these assets. Since the trend is towards more communication is growing, we expect the vulnerabilities to grow proportionally. This needs to be tackled using security assurance cases, especially by quality assurance which ensures the completeness of the case. It is also interesting that the assets within the hardware category are not higher than any of the assets in the software category. Such assets have been used in vehicles for quite a while, and it would make sense that the vulnerabilities have lowered in that category over time.

In Figure 5.1, the data storage category has the same amount of vulnerabilities for each asset. This makes sense because all the assets are related to storing data and

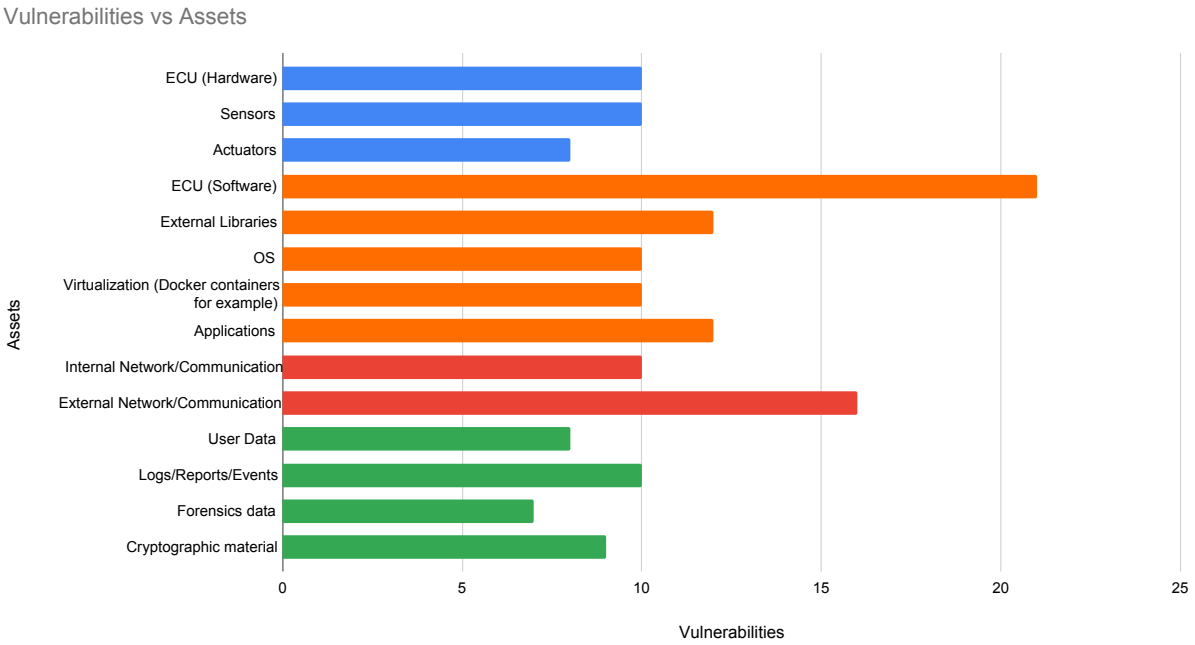


Figure 6.1: Chart of vulnerability count against assets.

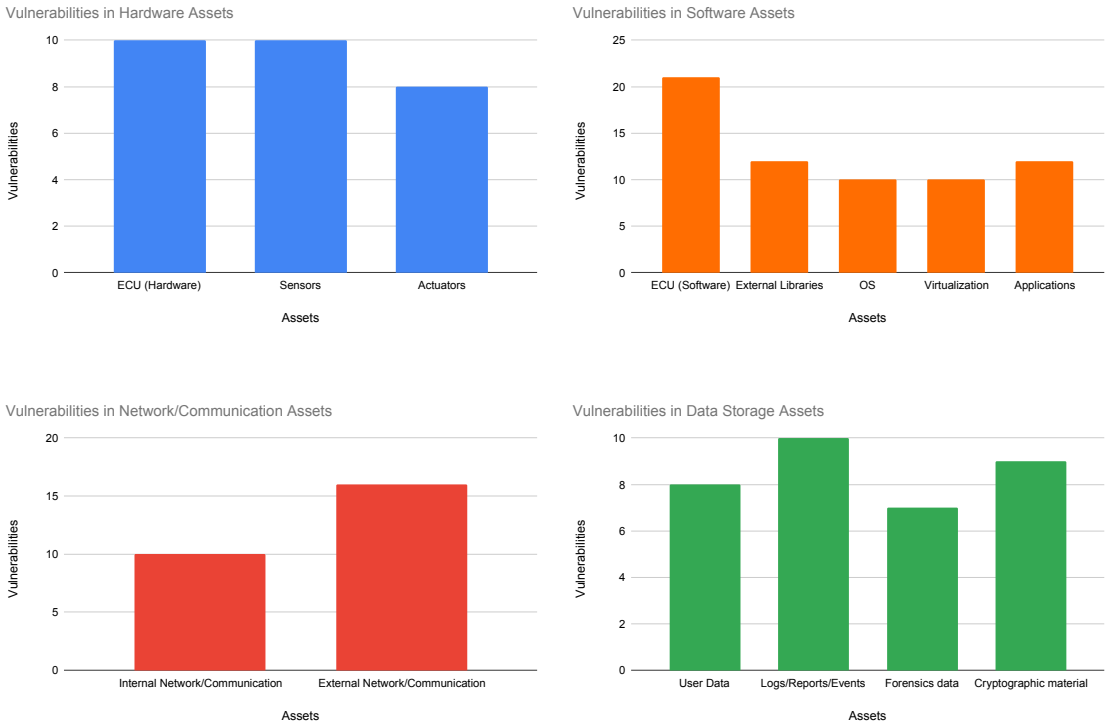


Figure 6.2: Vulnerability charts for each category

if there is any vulnerability that gets exploited by an attacker then all the sensitive assets in the data storage get affected which could lead to a data breach and/or loss

of data.

The revised version of the catalogue (Figure 5.4) shows that vulnerability references 5 and 9 are both related to "Threats to vehicles regarding their communication channels", as described in the GRVA [7]. According to our study and the mapping that we have conducted, almost all of the assets get affected by vulnerability 5 and nearly just as much by vulnerability 9. Both of these vulnerabilities are related to unauthorized access and will severely affect the CIA properties of the assets if attacked. The consequences of the success of such an attack include manipulation of vehicle data, erasing the vehicle data and code, and more. Thus, the catalogue indicates that these attacks are vital to prevent, and recommends mitigation strategies to implement.

The filters in the Types of Evidence catalogue were applied in order to find the types that would be of most use in an Automotive Development Process, focusing mainly on providing evidence for security within the respective categories. For this reason, the Actor category yielded only two types: Competence and Trustworthiness. These two types are important in this context since we want to make sure that the actors involved in the Automotive Development Process are competent enough to do their tasks with minimum mistakes, while having complete trust in them being part of the process. For example, developers with the task to implement security measures for authenticating users are able to do the task and do not have malicious intents.

Our goal for this research question was to find a way to support practitioners during the creation of Security Assurance Cases. We researched which assets, vulnerabilities, and controls would be able to support the creation of SACs, as well as which evidence exists that could support the claims of arguments. While we could not fully answer the latter, we did manage to find an alternative by identifying the types of evidence that could be used. By creating the catalogues, we identified assets, vulnerabilities, controls and types of evidence, all of which are needed and used in a Security Assurance Case. These catalogues can support the creation of SACs both during the creation process (as a verification tool) and post creation (as a quality assurance tool).

6.1.2 RQ2

The SAC that was created for the headlamp example was not extensive. As seen in figures in Appendix 2 (section B.1), the only block that was done fully was the Asset Identification & Decomposition block (Figure B.4). For this first example, we put a set amount of time for each section before moving on to the next, partly due to the time constraints. However, during this time we also wrote down the difficulties that were met. We found that the white-hat block was difficult to create since assets identified in the CASCADE example were different to the assets that we had identified in the catalogue. Additionally, the headlamp example in SystemWeaver had different columns for affected assets and requirements. The combination of both these columns had to be considered for the decomposition of assets. This made it

difficult to create security goals for decomposed assets.

During the creation of SACs, we consulted our first catalogue to compare the assets, vulnerabilities, and controls with those found in the examples. In a real life scenario, identifying missing artefacts for the SAC would result in reporting back to those responsible, or trying to somehow fix the issue. In our case, we could only take note of anything we found as it was just an example. The catalogue did prove to be quite supportive when it came to quality assurance since we could assess which assets, vulnerabilities, and controls were considered, both in the TARA and SAC, and which ones were not. We were also able to have short discussions about which types of evidence would be suitable for the case by using the second catalogue (types of evidence).

Although only two gaps were identified, they were still important. We confirmed, to some degree, that not only Systemite, but possibly other automotive companies have yet to **adopt the creation of SACs**. We also saw the risk-based activities (such as TARA) that the industry conduct for their systems, and they should now need to consider **how to use the outcomes of those activities to create security arguments for SACs** as they move towards adopting the creation of SACs. In addition to those findings, there were other things that we found that are important to mention.

For future practitioners of SACs, more examples of approaches would be very helpful so that the industry can at least develop a base for how to represent SACs. However, we do understand that each OEM might want to represent them in their own way. Before creating SACs, it is important that some preparation is done by gathering the documents you want to use and structuring them so that it would be easier for you to focus on the creation. We thought that one source, the TARA security grid (figure 5.14), would be enough for creating the C-ACC system Security Assurance Case, but we ended up using 4 sources in total: the security grid and attack event grid for both TARA level 1 and level 2. In order for the creation process to be a smooth experience, how you represent the data you have and how efficiently you can identify the elements you need for the SAC is important. Of course, this also depends on what approach you use. For an asset-based approach, such as CASCADE, you may prefer to have data represented in the same structure as the approach. A risk-based approach may be different. Either way, good preparation will most likely result in easier construction of the SAC.

For the second research question, the aim was to study whether the artifacts found in an Automotive Development Process would cover the needs in approaches found in the literature. During this time we would identify and analyse gaps that we found. For this purpose, we chose an approach and created two Security Assurance Cases. The chosen approach was CASCADE, which is an asset-driven approach. Using this approach we created two SACs, one regarding a Headlamp item and the other about a Cooperative Adaptive Cruise Control (C-ACC) system. We were met with many hurdles along the way which we addressed and successfully identified two

gaps. The first gap was hinted during our research into the first research question while the other one was identified after we had created the SACs. We managed to get into contact with a person outside of Systemite and sent them a questionnaire that helped validate the identified gaps. In addition to that, we also implemented a prototype of an SAC report template. This will act as a starting point for the company to create and present SACs for their customers.

6.2 Threats to Validity

To discuss threats to validity in this study, we consulted the book **Experimentation in software engineering** by Wohlin et al. [32]. The types of threats are: Internal validity, Construct validity and External validity.

6.2.1 Internal Validity

Cybersecurity, in recent years, has become an import aspect in the automotive industry, as well as others. Therefore, only recent papers (up to 3 years back) were studied in order to create the catalogues and build a Security Assurance Case based on an existing approach that is up to date. There is a risk that during the first iteration, additional relevant studies were not found in the research. The authors snowballed on older studies but not as thoroughly. There is a risk that the selected approach for building SACs might not be feasible for requirements of other companies working towards the creation of SACs.

Another threat to validity is the selection of the subject-matter experts that took part in the evaluation of results. The evaluation was performed with employees from Systemite, where some of them were experts in the subject of Cybersecurity and Assurance Cases. However, we do not know to what extent they have experience in the creation of Security Assurance Cases. Additionally, there is an uncertainty about the evaluation of the results since they came from the case company only. Furthermore, there might have been a slight conflict of interest in terms of what the outcome of the thesis would yield. While we set out to find answers to the questions we presented, there is a risk that the company's interest was, to simply put, have an upper hand in the market. This affects how we interpreted their evaluations and the impact of our contributions to this research topic.

6.2.2 Construct Validity

We would have liked to follow up on the evaluation done on the first catalogue on assets, vulnerabilities, and controls. Specifically the mapping done between assets and vulnerabilities, which were from two different sources of knowledge. Although there were separate analyses done which were later merged into the first version of the catalogue, a focus group with the aim to evaluate the mapping would increase the value of the catalogue.

6.2.3 External Validity

An attempt to generalise the results was made. To generalize the results, we would need to get in contact with other OEMs that would be interested in the topic. However, due to the time period in which this research was conducted, it was difficult to get in contact with relevant people. We did manage to get into contact with one person, but that only resulted in a questionnaire due to the time constraints. To mitigate this and make the results more generalised, future work would be to further evaluate the results with other companies in the automotive industry.

Another threat to validity is the gap analysis, where it was discovered that the approaches suggested in literature should be modified based on Systemite's requirements. Evaluating Security Assurance Cases may take time as most of the companies in the automotive industry have not fully adapted to implementing SACs. However, the catalogue which are created for the support of SACs can be evaluated with other companies to further generalise the results.

7
Conclusion & Future Work

In this section we will conclude our work and discuss possible future work that can be done following the results of this thesis.

7.1 Conclusion

The development towards connected cars technology has been very innovative but not so much in terms of cybersecurity. With services that require internet connection and being a part of the Internet of Things (IoT), the vehicles become more vulnerable to cyber attacks, such as taking over the control of the steering. The ISO/SAE-21434 standard (under development) will demand automotive vehicles to ensure the security in the automotive vehicles that they manufacture. This can be achieved by Security Assurance Cases (SACs), although the automotive industry has yet to adopt it as part of their Automotive Development Process.

In order to assist the adoption and support the creation of SACs in the automotive industry, two catalogues were made. One identifying the assets, vulnerabilities, and controls in a ADP, and the other providing a catalogue of the *types* of evidence that can be used to support the arguments of the claims in a SAC. Further more, a gap analysis was conducted that would address the disproportion of research of SACs in relation to the application of SACS in the automotive industry. This was done in collaboration of our case company Systemite, and their system engineering tool, SystemWeaver.

The mapping of the identified assets, vulnerabilities, and controls for the first catalogue gave interesting insight about the relationship between the assets and cybersecurity. The vulnerabilities vs asset chart provided insight into degree of vulnerability of each asset, to a certain extent. To no surprise, the software related Electronic Control Units (ECUs) were found to be most vulnerable with external network/communication assets being not far behind. Aside from the positive evaluations, the two catalogues that were implemented into SystemWeaver brought a lot interest to other employees at our case company. Furthermore, the catalogues were purposely made open-source as our intentions were to contribute our findings to all practitioners of SACs, both present and future. As we used the catalogues during the second part of our research, we saw a lot of potential in them.

The gap analysis conducted resulted in two SACs by following the CASCADE approach. Despite the intuitive way of how to structure your SAC, the approach was

not easy to use. Working with only the headlamp example from the CASCADE paper, it was difficult to apply the same steps on another system or even on the same system but with different assets. This goes to show that many more example will be needed in order to become well versed with creating SACs.

While the automotive industry has yet to adopt the creation of SACS (gap one), the second gap might be one answer as to how to proceed from there. Since the adoption of Safety in the automotive industry, Threat Analysis & Risk Assessment has been a popular risk-based activity. The question now is: how can we use the outcome of those activities to create security arguments for SACs (gap two). Finding an answer to this question will definitely improve the transition from Safety to Security, as well as save time and money by using existing artifacts in an ADP.

Finally, the SAC Report template implemented in SystemWeaver provides a suggestion of sample security case, both for our case company and other automotive companies that will be creating SACs. The structure of the report is built in a way that provides both the SAC and evidence for the SAC. The report makes use of the second catalogue where it shows all the types of the evidence and their description in order to provide as much information as possible in the document. Although the report is just a template/prototype, the case company can further refine it and use it towards their customers.

7.2 Future work

For future work, we think further evaluation of the results should be conducted by subject matter experts outside of Systemite in order to provide more generalizability. The catalogues can be evaluated by other parties to assess the usability and quality of the results, and the gaps can be further investigated. Additionally, the catalogues can be improved by identifying more assets and evidence and expand on the tables that currently exist.

The second gap is an important finding that should be further investigated. Risk-based activities are important, if not just as much, in security as they are in safety. However, they are slightly different in their respective fields. Therefore, future research could be to investigate the second gap. Namely, how to use the outcome of the risk-based activities as security arguments in SACs.

Last but not least, the development of the SAC report template into a version 1 could be performed. This could result in a standardized report that the automotive industry could use for their SACs, or derive their own versions from.

Bibliography

[1] R. Coppola and M. Morisio, "Connected car: technologies, issues, future trends," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, pp. 1–36, 2016.

[2] R. K. Bajaj, M. Rao, and H. Agrawal, "Internet of things (iot) in the smart automotive sector: a review,"

[3] G. Bella, P. Biondi, G. Costantino, I. Matteucci, and M. Marchetti, "Towards the cosca framework for "conseptualing secure cars".," *Open Identity Summit 2021*, 2021.

[4] M. Mohamad, J.-P. Steghöfer, and R. Scandariato, "Security assurance cases—state of the art of an emerging approach," *Empirical Software Engineering*, vol. 26, no. 4, pp. 1–43, 2021.

[5] "ISO 26262 road vehicles — functional safety." `https://www.iso.org/standard/68383.html`. (accessed: 2021-08-10).

[6] "ISO/SAE 21434 Road vehicles — Cybersecurity engineering." `https://www.iso.org/standard/70918.html`. (accessed: 2021-08-10).

[7] W. UNECE, "Grva, "draft recommendation on cyber security of the task force on cyber security and over-the-air issues of unece wp. 29 grva."," 29.

[8] N. Mansourov and D. Campara, *System assurance: beyond detecting vulnerabilities*. Elsevier, 2010.

[9] Systemite AB. `https://www.systemweaver.se/`. (accessed: 2021-08-19).

[10] A. Finnegan and F. McCaffery, "Towards an international security case framework for networked medical devices," in *International Conference on Computer Safety, Reliability, and Security*, pp. 197–209, Springer, 2014.

[11] Systemite AB. `https://www.linkedin.com/company/systemite/about/`. (accessed: 2021-08-19).

[12] "ISO - International Organization for Standardization." `https://www.iso.org/about-us.html`. (accessed: 2021-08-10).

[13] A. C. W. Group *et al.*, *Goal structuring notation community standard (version 2)*. Jan, 2018.

[14] C. B. Weinstock, H. F. Lipson, and J. Goodenough, "Arguing security-creating security assurance cases," in *tech. rep., Software Engineering Institute*, 2007.

[15] S. C. S. Club, "Goal structuring notation." `https://scsc.uk/gsn`. (accessed: 2021-08-18).

[16] J. Spriggs, *GSN - the goal structuring notation: A structured approach to presenting arguments*. Springer Science & Business Media, 2012.

[17] M. Mohamad, A. Åström, Ö. Askerdal, J. Borg, and R. Scandariato, "Security assurance cases for road vehicles: an industry perspective," in *Proceedings of the*

15th International Conference on Availability, Reliability and Security, pp. 1–6, 2020.

[18] V. Patu and S. Yamamoto, "How to develop security case by combining real life security experiences (evidence) with d-case," *Procedia Computer Science*, vol. 22, pp. 954–959, 2013.

[19] K. Netkachova, K. Müller, M. Paulitsch, and R. Bloomfield, "Investigation into a layered approach to architecting security-informed safety cases," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, pp. 6B4–1, IEEE, 2015.

[20] J. Weber, *Automotive development processes: Processes for successful customer oriented vehicle development*. Springer Science & Business Media, 2009.

[21] M. S. U. Alam, S. Iqbal, M. Zulkernine, and C. Liem, "Securing vehicle ecu communications and stored data," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2019.

[22] T. Rosenstatter, K. Strandberg, R. Jolak, R. Scandariato, and T. Olovsson, "Remind: A framework for the resilient design of automotive systems," in *2020 IEEE Secure Development (SecDev)*, pp. 81–95, IEEE, 2020.

[23] O. Burkacky *et al.*, "Cybersecurity in automotive," tech. rep., McKinsey, 2020.

[24] D. Walkowski, "What is the CIA Triad?." `https://www.f5.com/labs/articles/education/what-is-the-cia-triad`. (accessed: 2021-08-19).

[25] F. Cyber Edu, "What is the CIA Triad? Defined, Explained, and Explored." `https://www.forcepoint.com/cyber-edu/cia-triad`. (accessed: 2021-08-19).

[26] L. B. Othmane and A. Ali, "Towards effective security assurance for incremental software development the case of zen cart application," in *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pp. 564–571, IEEE, 2016.

[27] "Diagram software and flowchart maker." `https://www.diagrams.net/`. (accessed: 2021-08-13).

[28] V. Vaishnavi, W. Kuechler, and S. Petter, "Design science research in information systems," *January*, vol. 20, p. 2004, 2004.

[29] M. Mohamad, Ö. Askerdal, R. Jolak, J.-P. Steghöfer, and R. Scandariato, "Asset-driven security assurance cases with built-in quality assurance," in *2021 IEEE/ACM 2nd International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS)*, pp. 1–8, IEEE, 2021.

[30] C. B. Weinstock and H. F. Lipson, "Evidence of assurance: laying the foundation for a credible security case," tech. rep., Carnegie Mellon University Pittsburgh United States, 2013.

[31] Google, "Google sheets." `https://www.google.com/sheets/about/`. (accessed: 2021-08-19).

[32] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

A
Appendix 1

A.1 Type of Evidence - GSN representations

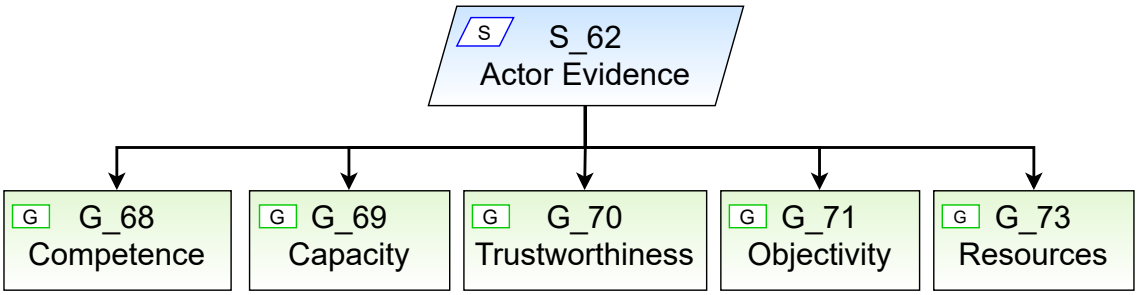


Figure A.1: GSN graph representation of actor evidence.

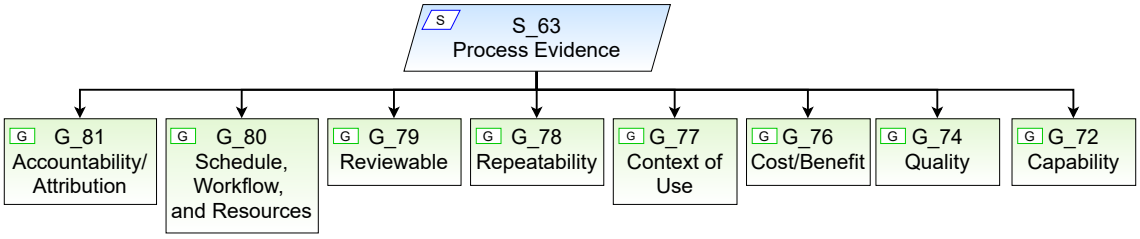


Figure A.2: GSN graph representation of process evidence.

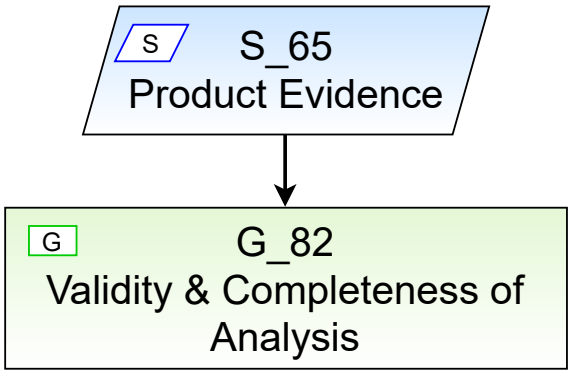


Figure A.3: GSN graph representation of product evidence.

B
Appendix 2

B.1 Headlamp SAC

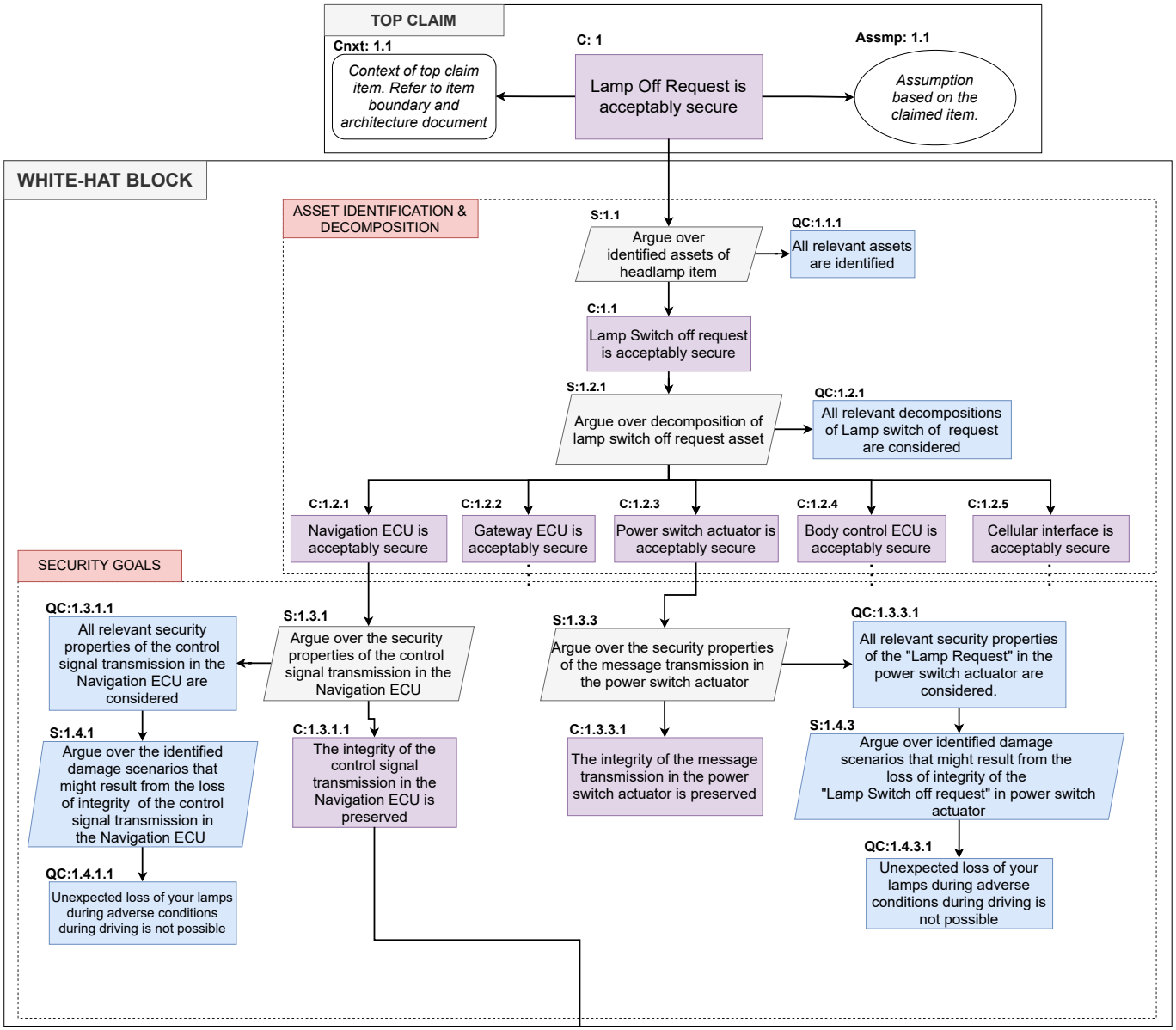


Figure B.1: The top-level claim and the white-hat block for the headlamp item.

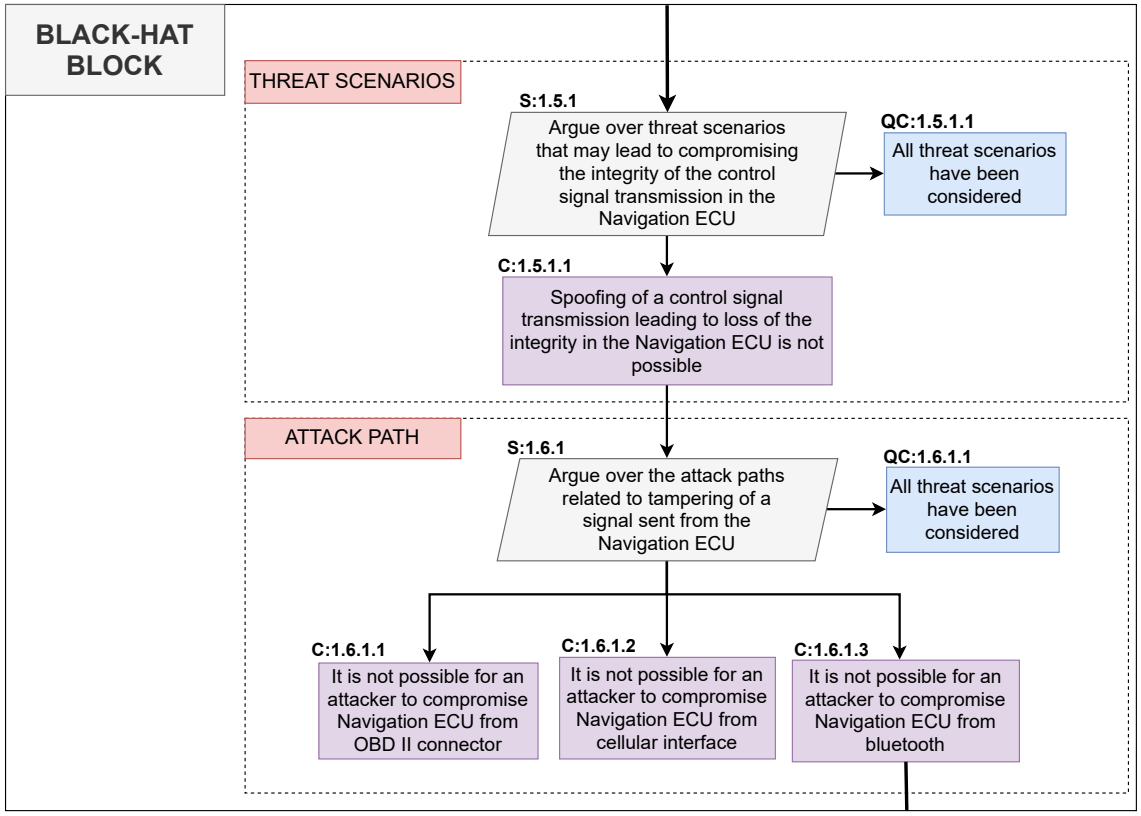


Figure B.2: The black-hat block for the headlamp item which includes claiming the negation of the damage scenarios and attack paths.

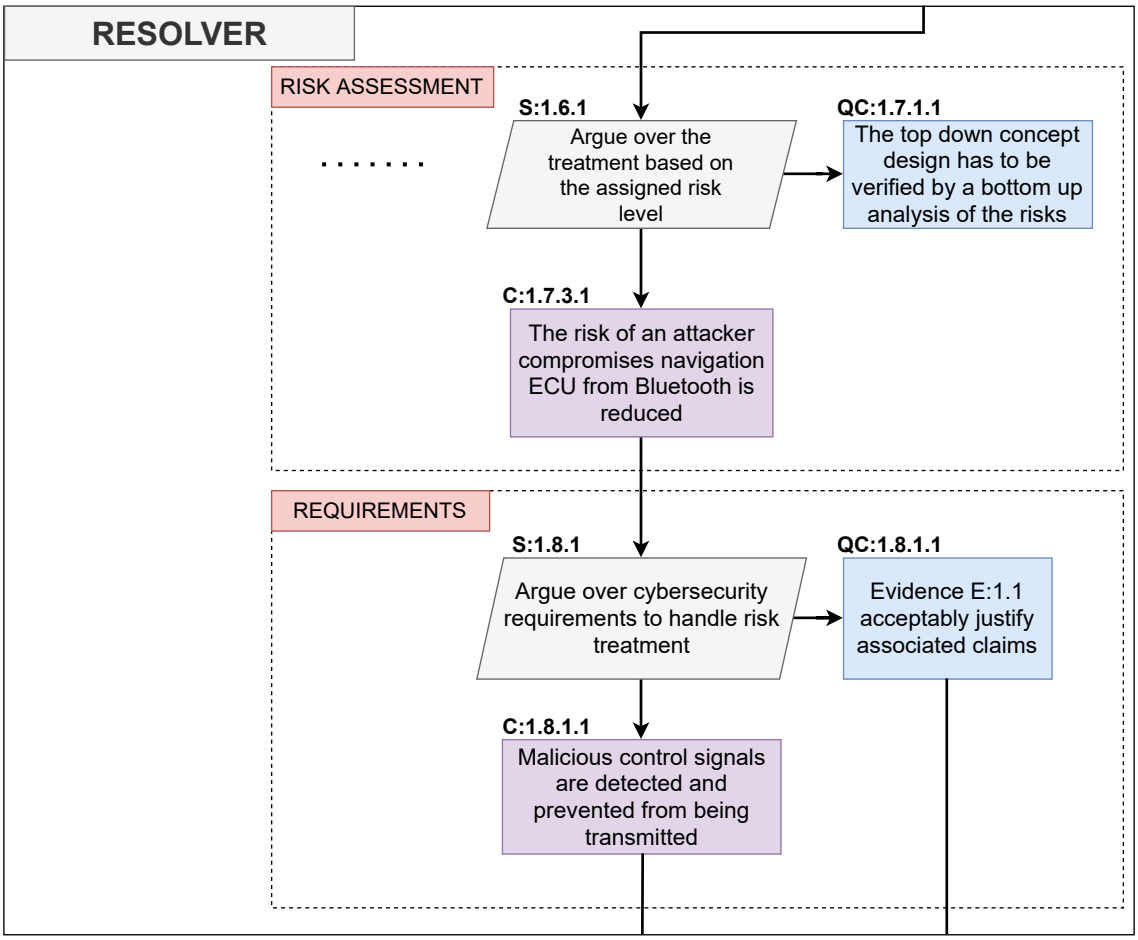


Figure B.3: The resolver block for the headlamp item.

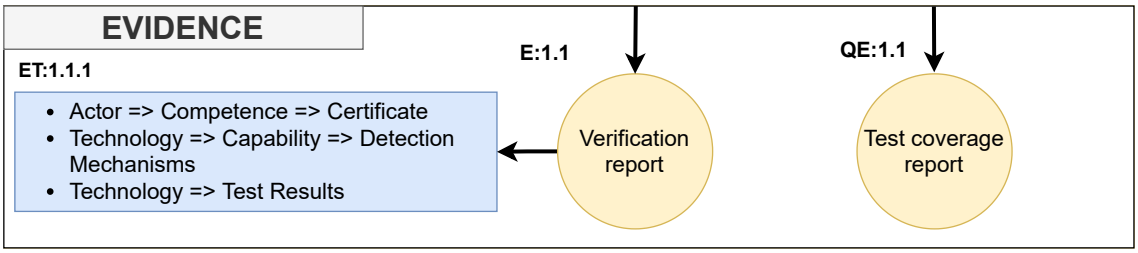


Figure B.4: The evidence block for the headlamp item. Note that real evidence did not exist for the given example, thus placeholder names were used. Additionally, types of evidence was used to provide aid the selection of evidence needed for the claims.

B.2 C-ACC SAC Systemite Redrawn

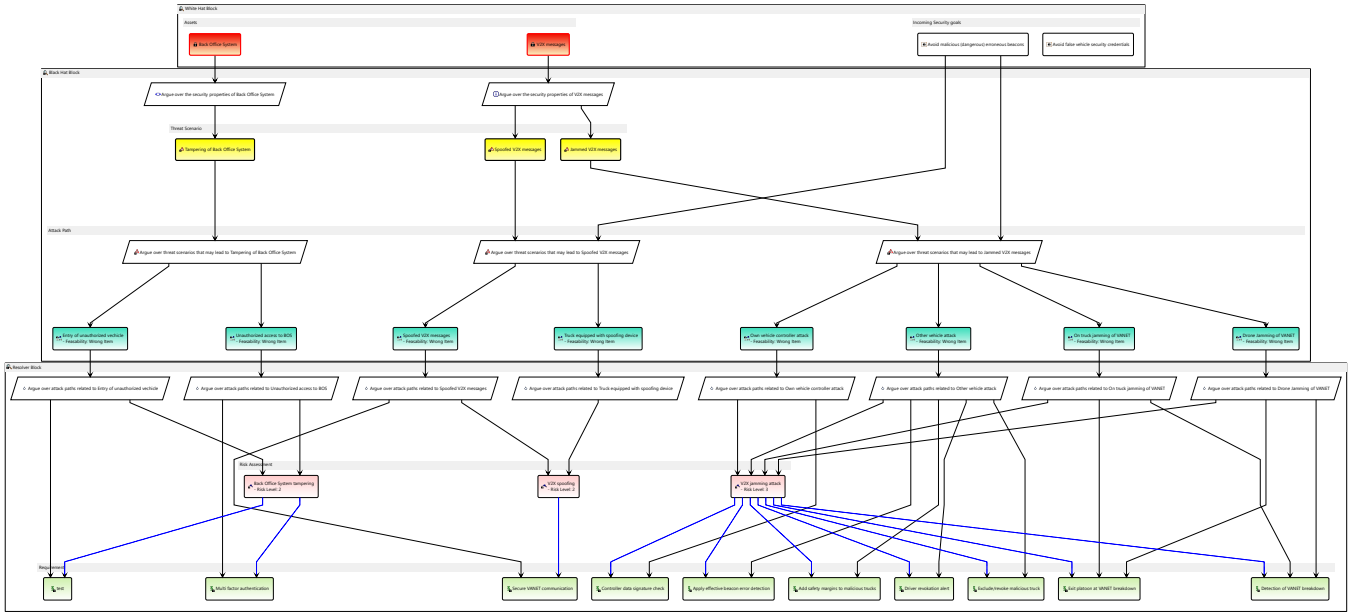


Figure B.5: The original SAC graph that SystemWeaver produced of the C-ACC system.

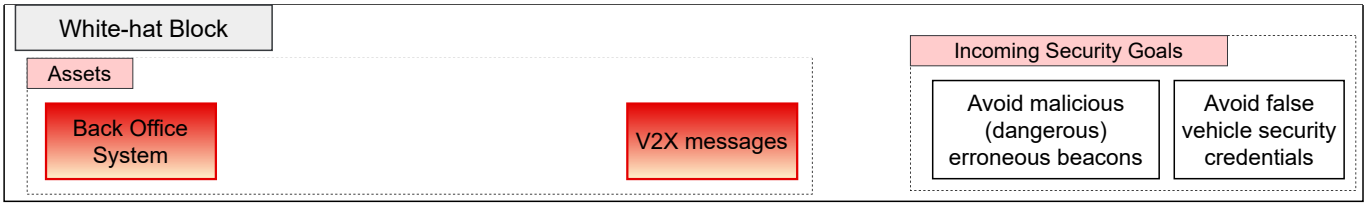


Figure B.6: White-hat Block of the redrawing.

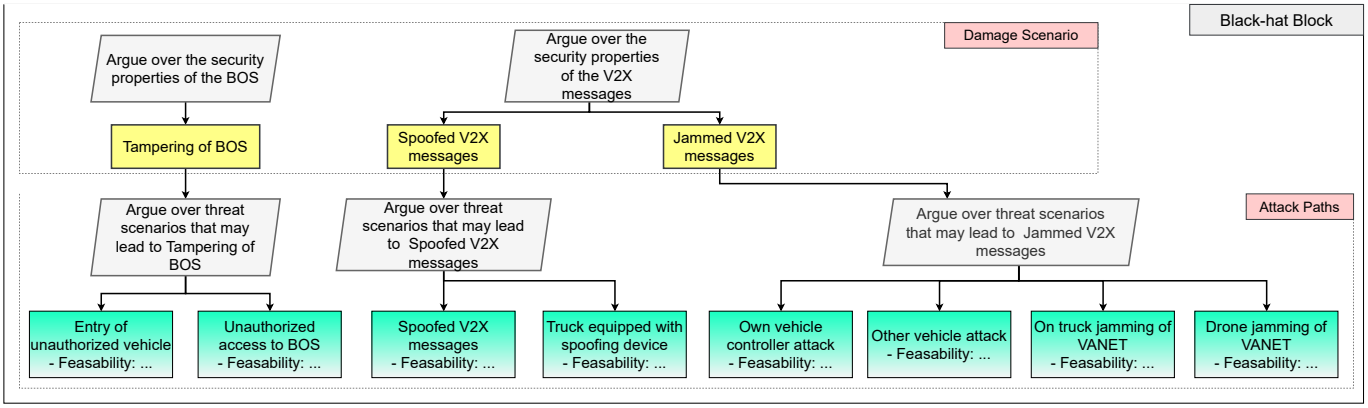


Figure B.7: Black-hat Block of the redrawing.

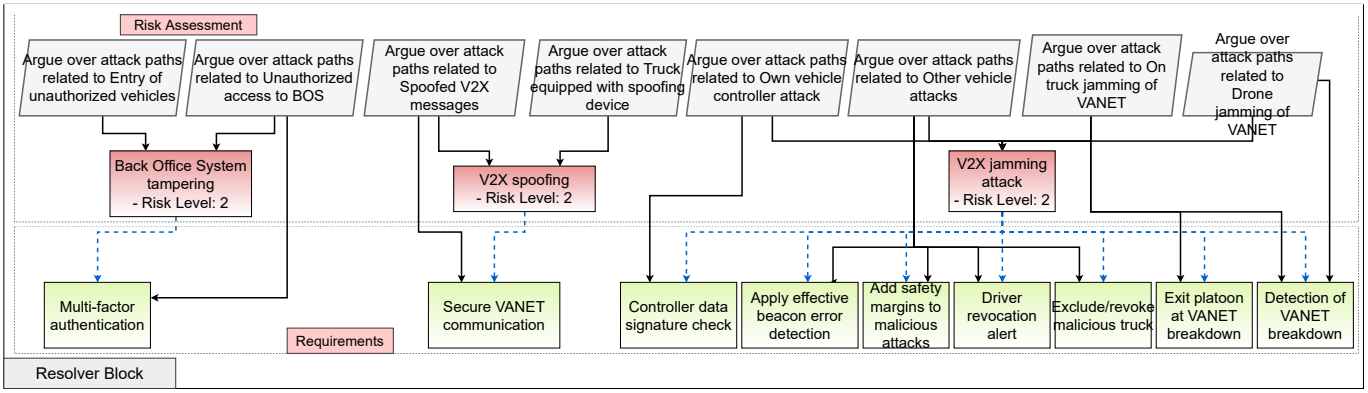


Figure B.8: Resolver Block of the redrawing

B.3 C-ACC SAC

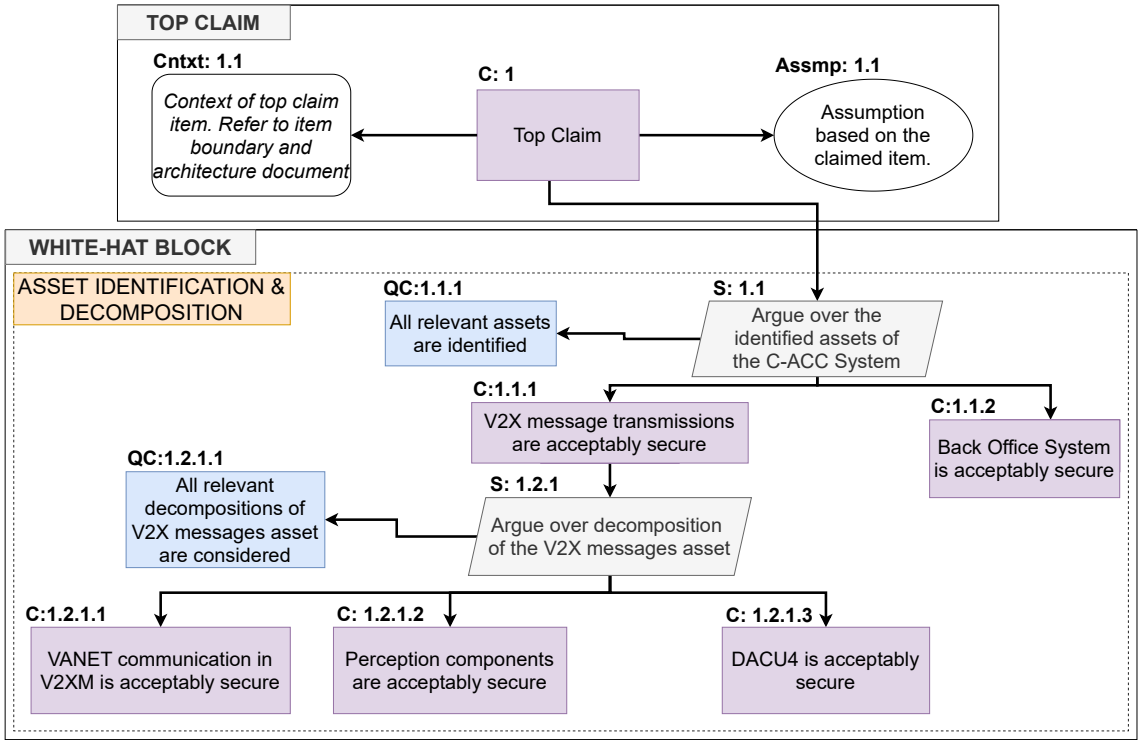


Figure B.9: The Top Claim and Asset Identification & Decomposition of the White-hat block for the C-ACC Security Assurance Case. See Figures B.10, B.11, B.12, and B.13 for the individual paths. Figure B.14 shows the Resolver and Evidence Block for the case.

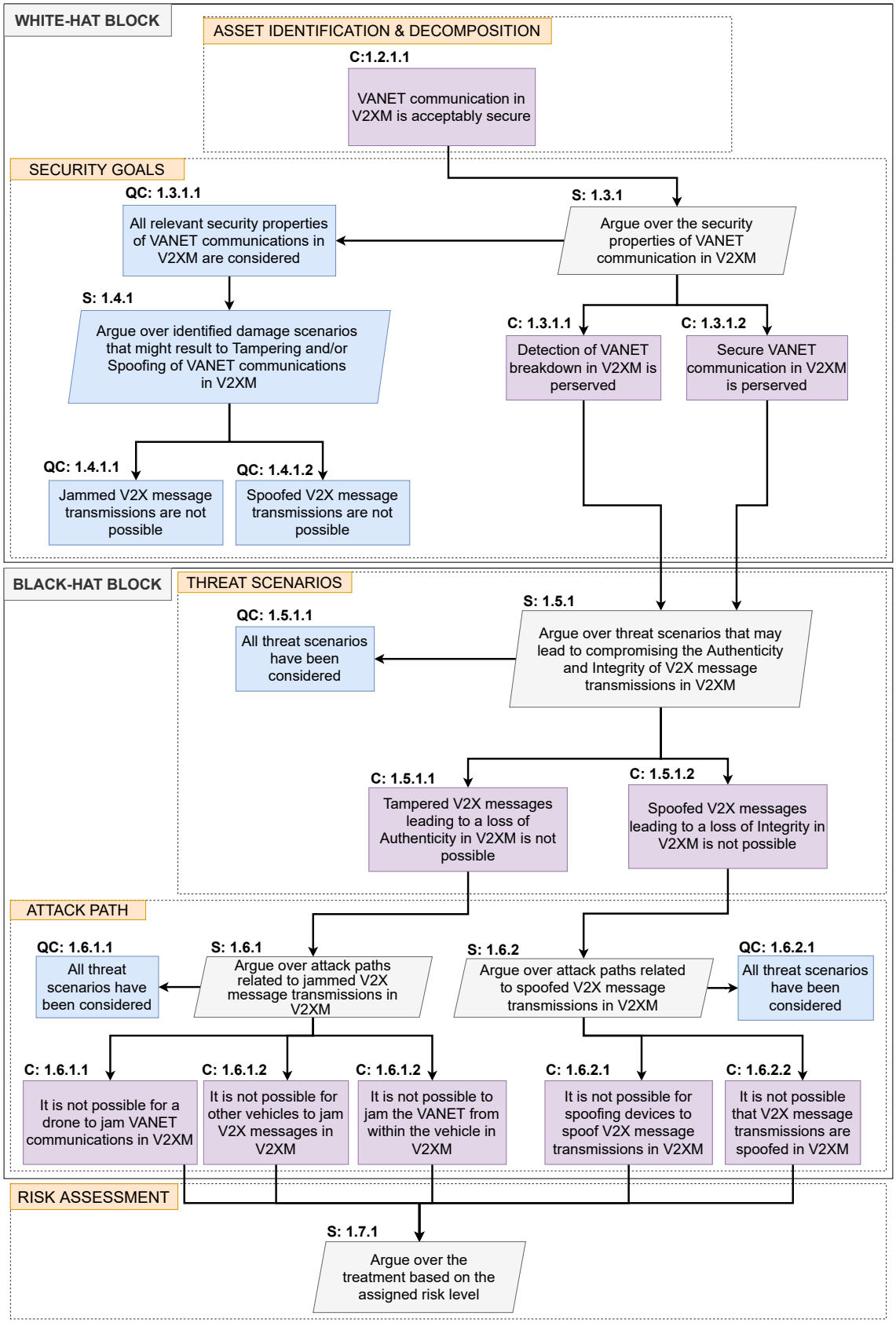


Figure B.10: The first path down the V2X decomposition. Starts from the decomposed asset and goes down to Risk Assessment. All claims from Attack Path lead to strategy S:1.7.1, shown in figure B.14.

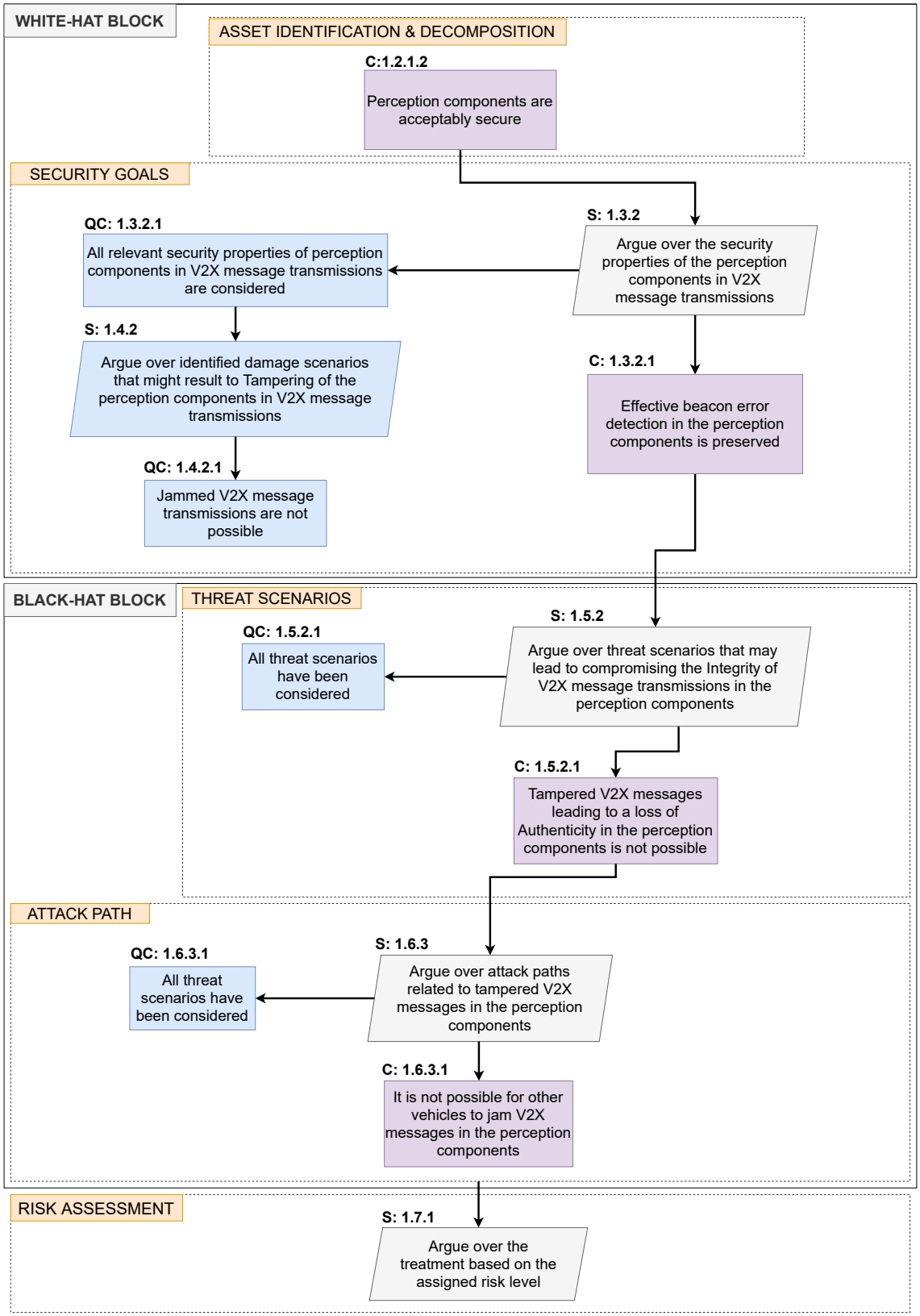


Figure B.11: The second path down the V2X decomposition. Starts from the decomposed asset and goes down to Risk Assessment. All claims from Attack Path lead to strategy S:1.7.1, shown in figure B.14.

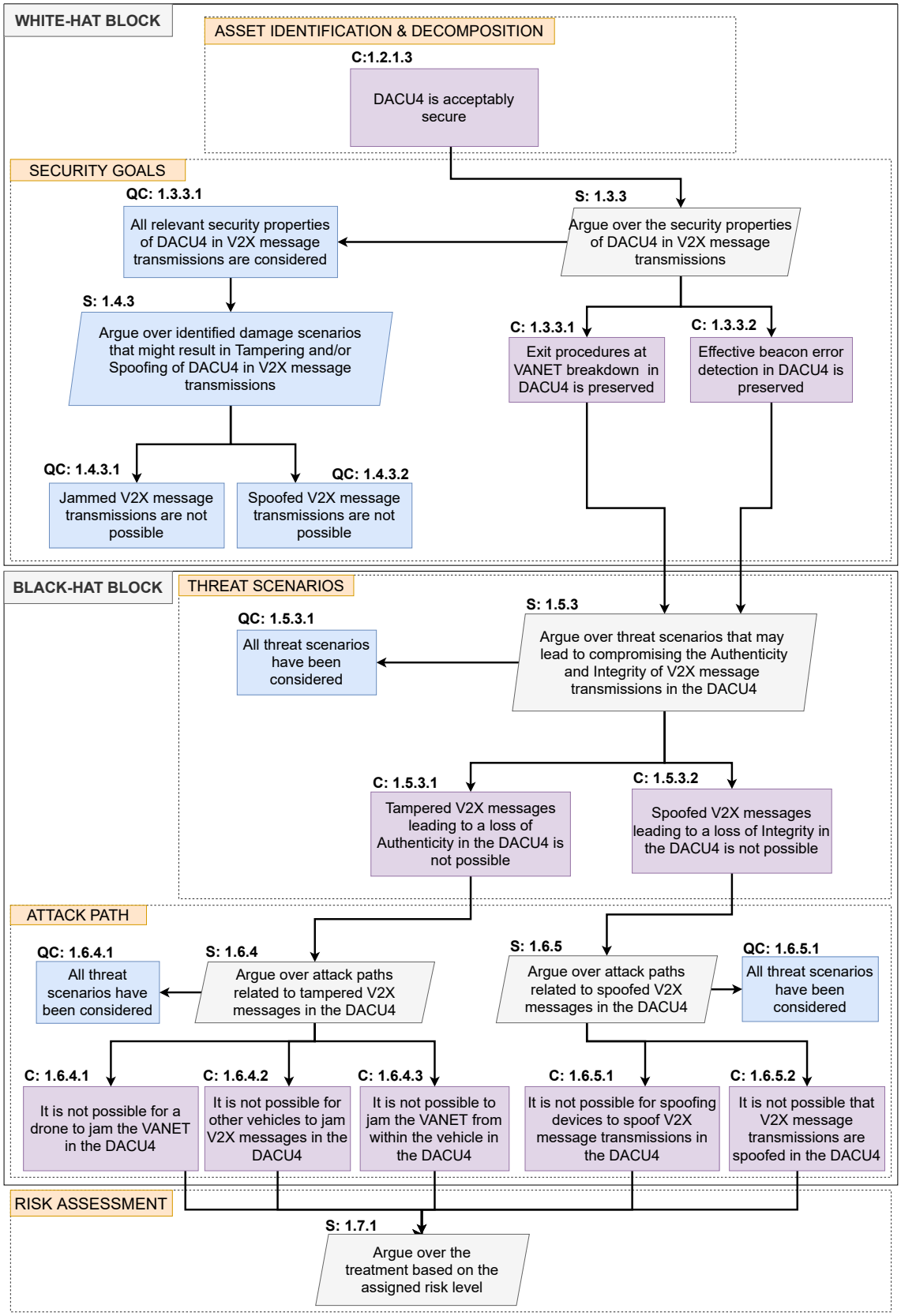


Figure B.12: The third path down the V2X decomposition. Starts from the decomposed asset and goes down to Risk Assessment. All claims from Attack Path lead to strategy S:1.7.1, shown in figure B.14.

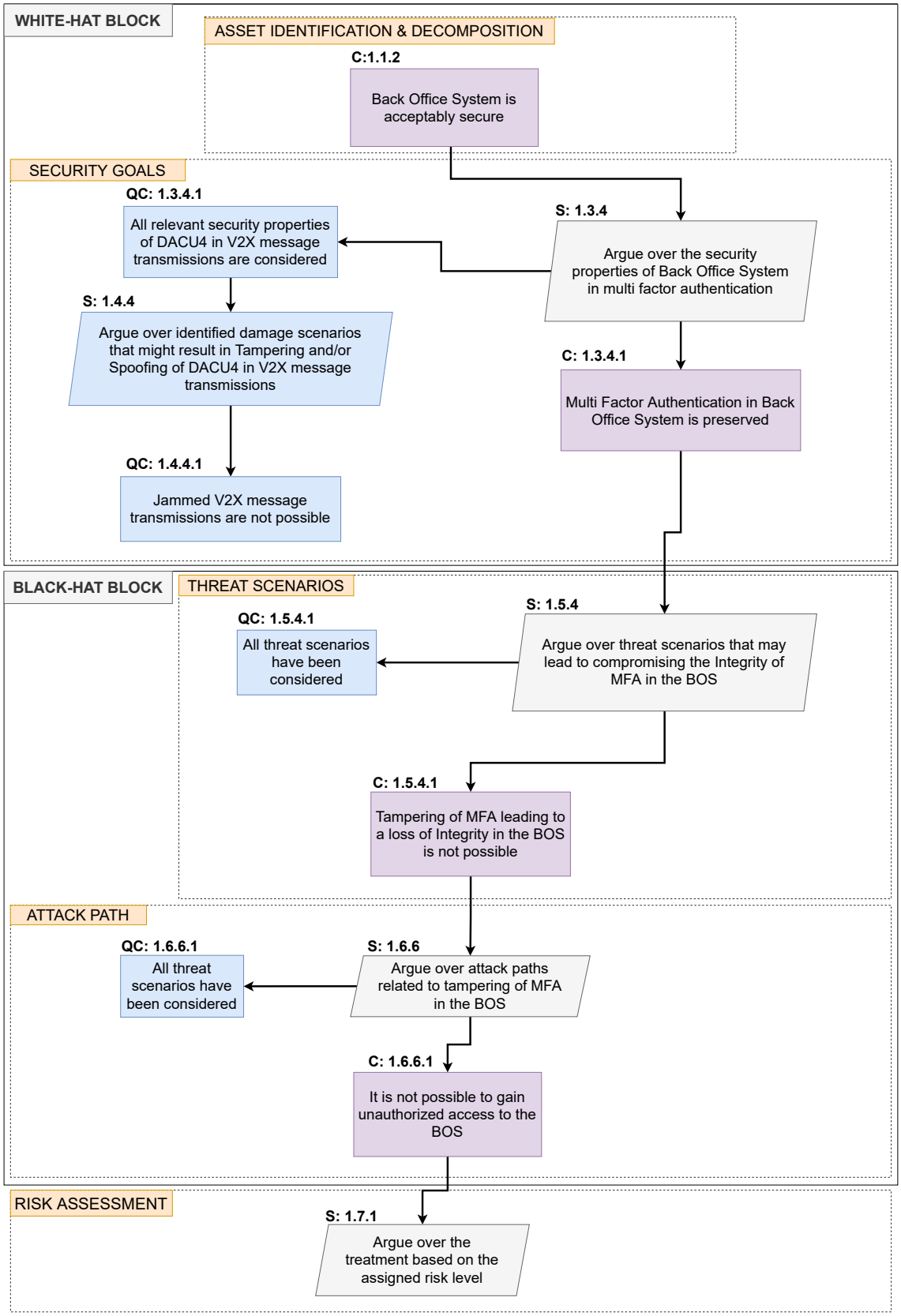


Figure B.13: The path down the Back Office System. Starts from the asset identification & decomposition and goes down to Risk Assessment. All claims from Attack Path lead to strategy S:1.7.1 in Risk Assessment, shown in figure B.14.

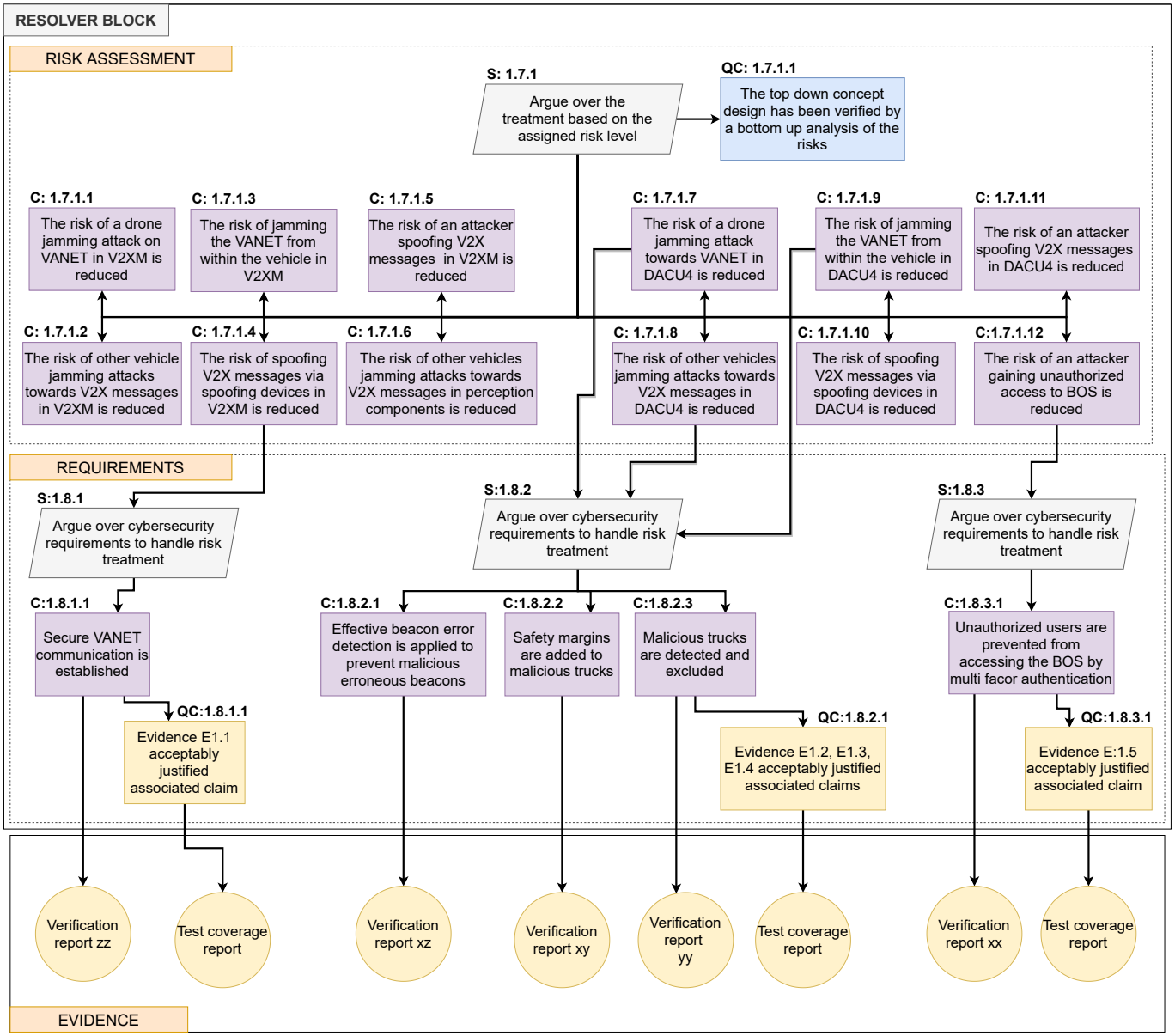


Figure B.14: Resolver Block (Risk Assessment and Requirements) and Evidence block of the C-ACC Security Assurance Case.

B.4 SAC Report Template

1 Introduction

This document provides structured argumentation regarding the security of an item. Further down the document you will see a Security Assurance Case of the item. This SAC will provide what argumentation strategies for the claims/goals that fulfill the security goals/requirements, as well as evidence to support those claims/goals.

2 Background

This section aims to provide further information about what you will be seeing in this document.

2.1 Security Assurance Case

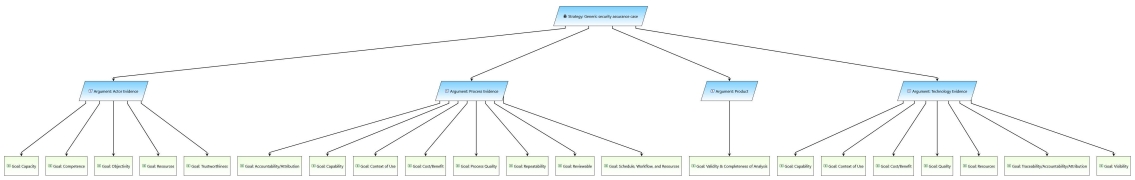
Assurance cases are a collection of evidences brought forth in structured arguments. They are used to argue that a particular claim about a systems property holds.

In the case of a Security Assurance Case (SAC), the claims are about the security aspect about a system where the evidence is used to strengthen the aforementioned claim. SACs can be created with the help of GSN. The GSN provides the following nodes in order to create a case: claim (also called goal), context, strategy, assumption (also called justification), and evidence (also called solution). The case consists of a high level claim at the top which is broken down into sub-claims based on certain strategies. The claims are used to specify the goals we want to assure in the case, e.g., a certain system/feature is secure. A strategy for instance can be broken down based on certain security attributes. The claims are broken down repeatedly till they reach a point where evidence can be assigned to justify the claims/sub-claims. Examples of evidence can be code review reports and test results. The assumptions can be made while applying the strategies.

3 Argumentation

3.1 Generic security assurance case

Claim diagram:

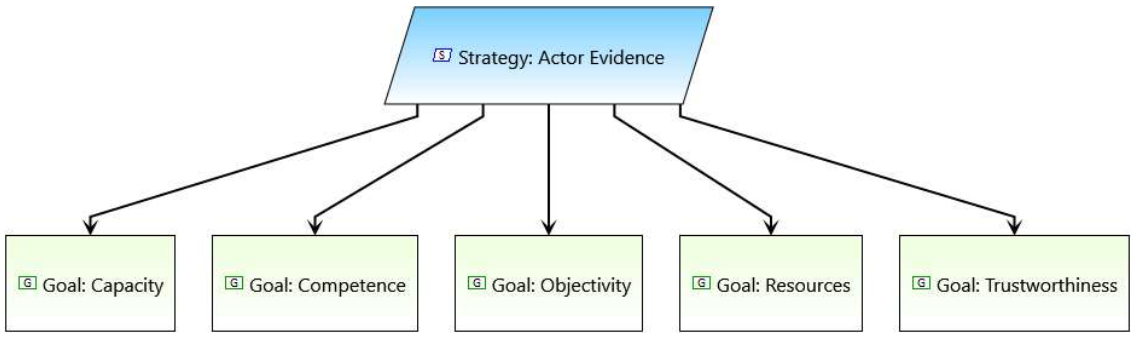


Claim table:

Initial Cybersecurity Case	Cybersecurity Case Text	Strategies	Goals
Generic security assurance case	The types and attributes of the evidence shown in this document are based on the paper "Evidence of Assurance: Laying the Foundation for a Credible Security Case". For more information and further examples, please consult the paper: https://us-cert.cisa.gov/bsi/articles/knowledge/assurance-cases/evidence-assurance-laying-foundation-credible-security-case	Actor Evidence	Capacity
			Competence
			Objectivity
			Resources
			Trustworthiness
		Process Evidence	Accountability/Attribution
			Capability
			Context of Use
			Cost/Benefit
			Process Quality
			Repeatability
			Reviewable
			Schedule, Workflow, and Resources
		Product	Validity & Completeness of Analysis
		Technology Evidence	Capability
			Context of Use
			Cost/Benefit
			Quality
			Resources
			Traceability/Accountability/Attribution
			Visibility

3.1.1 Actor Evidence

Claim diagram:



Strategy table:

Strategy Name	Strategy Description	Strategy Goals
Actor Evidence	A participant in the development process. Evidence for actors rely on their competence and performance ability among other aspects. An actor could be a developer, product manager, stakeholder, organization, etc.	Capacity
		Competence
		Objectivity
		Resources
		Trustworthiness

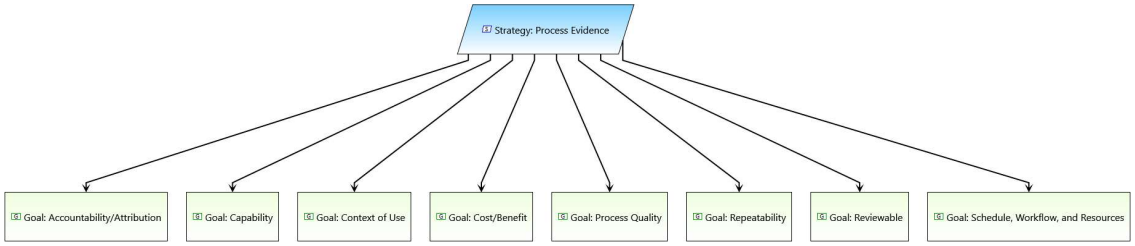
Goal table:

Goal Name	Goal Description
Capacity	How much can be accomplished within a given period of time? Example(s): • Report of an actors efficiency.

Goal Name	Goal Description
	• Reports of performance in previous projects.
Competence	An actor's skills/expertise for a given task, or more generally, in a specific domain (e.g., credentials are one source of evidence) Example(s): • Certificate Evidence (Training and Education) • Training process evidence
Objectivity	Absence of conflicts of interest in a given context Example(s): • Test Case Report
Resources	Assets (including economic assets of organization) Example(s): • Tangible Assets • Intangible Assets • Current Assets • Fixed Assets • Operating Assets • Non-operating Assets • Financial Assets
Trustworthiness	This relates solely to intent: the actor's veracity, honesty, and alignment with the mission of the system. Example(s): • Report of an actors integrity in previous projects • Report of actors credentials with respect to their experience

3.1.2 Process Evidence

Claim diagram:



Strategy table:

Strategy Name	Strategy Description	Strategy Goals
Process Evidence	A series of actions or steps taken in order to achieve a particular end. Evidence in this category should include reports/analysis/assessments/documents about particular processes, including (but not limited to) their capability, quality, cost/benefit, etc.	Accountability/Attribution Capability Context of Use Cost/Benefit Process Quality Repeatability Reviewable Schedule, Workflow, and

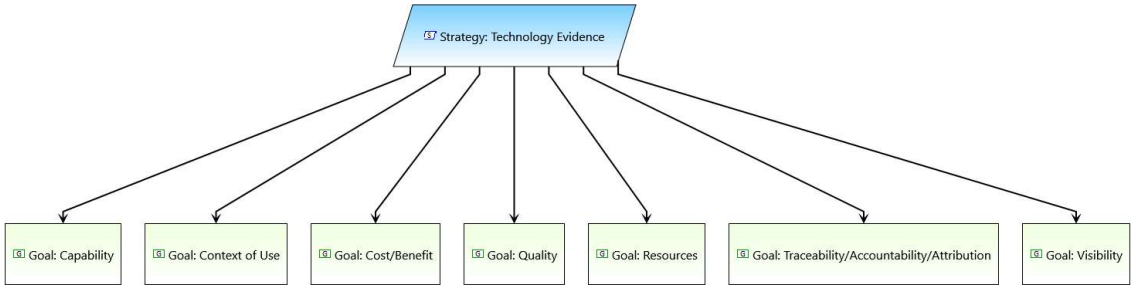
Strategy Name	Strategy Description	Strategy Goals
		Resources

Goal table:

Goal Name	Goal Description
Accountability/Attribution	Does the process keep track of the actions of the participating actors, and are the appropriate actors accountable for their actions (i.e., are the results of significant activities attributable to specific actors?) For example, are test results dated, linked to the correct version number of the code, digitally signed, and stored in a database for future use as evidence of assurance? Example(s): • Document showing step-by-step track of participating actors and their actions
Capability	What can actors accomplish by using the process? Example(s): • Test Case Report
Context of Use	What is the context in which the technology/process is applicable and achieves valid results? Example(s): • Use Case Report
Cost/Benefit	How practical is the process, i.e., how much does the process cost with respect to the value obtained? Example(s): • Cost/Benefit Analysis
Process Quality	How good is the process at achieving a desired result, be it analysis of code for a desired security property or the ability to construct a component with a high assurance of security? Quality arguments include conformance to best practices as embodied in recognized standards (or the more general claim of "adherence to industry standard practice" or "due care") and the existence of studies that validate the effectiveness of the process, as well as typically weaker arguments about the performance history associated with systems for which this process was used Example(s): • Test Report
Repeatability	Within its context of use, is the process readily repeatable over time across different project teams, across different organisations, even across different application domains (i.e., industry segments)? For example, is the process well documented and easy to follow? What organizational and individual resources are needed (e.g., skill sets and tools) so that the process produces the same results each time it is executed, regardless of who executes it? Example(s): • Experimentl Report • Modularity & Integration
Reviewable	Does the process document intermediate steps (intermediate inputs and outputs), along with associated rationale, so that the entire process is reviewable (for example, by an independent third party)? Example(s): • Document showing step-by-step guide, evaluated by other actors perhaps.
Schedule, Workflow, and Resources	What actor resources (with what skill sets) and what technology resources are required to carry out the process, and for how long are they needed? What interactions are there among the individuals and workgroups (internal and external) participating in the process? How predictable is the schedule for the process? This attribute is closely related to the cost/benefit attribute. Example(s): • Cost/Benefit Analysis

3.1.3 Technology Evidence

Claim diagram:



Strategy table:

Strategy Name	Strategy Description	Strategy Goals
Technology Evidence	The technology being used in the development process, for example, Electronic Control Units. Evidence in this category should include test reports, analysis, documentation, etc., that support that the technological assets are secure. Capability, quality, and traceability are some of the types of evidence to gather.	Capability
		Context of Use
		Cost/Benefit
		Quality
		Resources
		Traceability/Accountability/Attribution
		Visibility

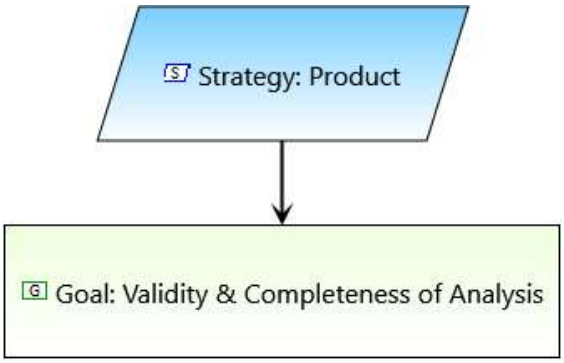
Goal table:

Goal Name	Goal Description
Capability	What can the technology accomplish? Example(s): Test Results
Context of Use	What is the context in which the technology is applicable and achieves valid results? Example(s): • Use Case Report • System Architecture • Item Boundary
Cost/Benefit	How affordable is the technology relative to its value? Example(s): • Cost/Benefit Analysis
Quality	Which software or system quality attributes (security, reliability, fault tolerance, etc.) are associated with the technology? Example(s): • Quality Attribute Analysis/Report
Resources	What actor resources (with what skill sets), processes, and other technology resources are required to make effective use of the technology? Example(s): • Actor, Process • Technology Assets
Traceability/Accountability/Attribution	Does the technology keep track of the actions of the participating actors, and are the results of significant activities attributable to specific actors? Are significant events logged and available for audit? Example(s):

Goal Name	Goal Description
	• Traceability of technology tasks/implementations to actors
Visibility	Are the artifacts of life cycle processes used to create the technology available to users, customers, certifiers and others, so that the technology's quality attributes can be assessed through analysis of those artifacts? Example(s): • Artifacts analysis document

3.1.4 Product

Claim diagram:



Strategy table:

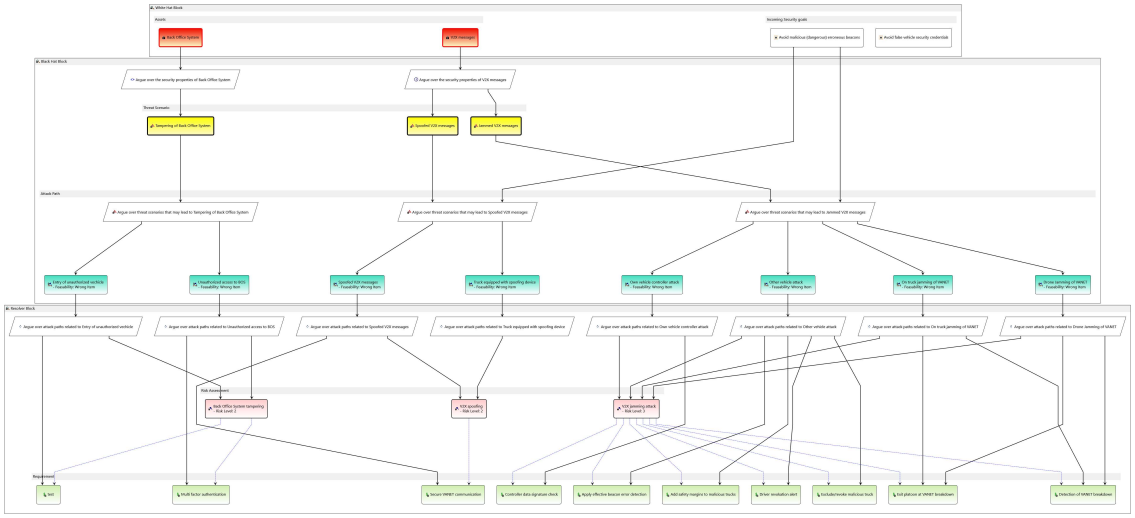
Strategy Name	Strategy Description	Strategy Goals
Product		Validity & Completeness of Analysis

Goal table:

Goal Name	Goal Description
Validity & Completeness of Analysis	

3.1.4.1 C-ACC TARA Level-2 Security Assurance Case

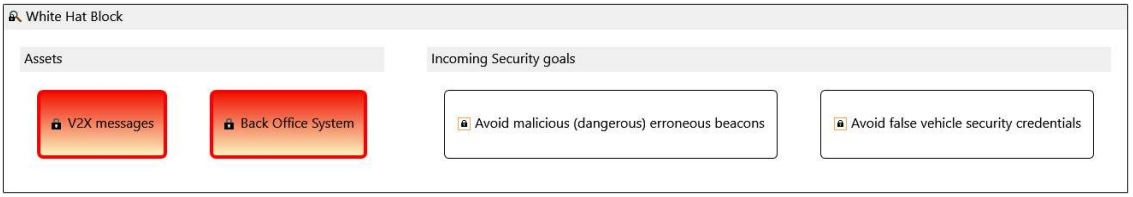
Full SAC Diagram



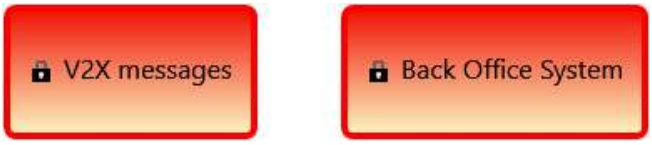
White hat

This block is about the assets and security goals. Asset identification is done by conducting an analysis to find the artefacts of the system that are likely to be subject to an attack. The security goals come from identifying relevant security properties for the assets. Most often you would look into the Confidentiality, Integrity, and Availability (CIA) triad.

White Hat Diagram



Assets



Asset Name	Asset Description	Asset Type
Back Office System		Security System Component
V2X messages		Interface

Incoming Security Goals

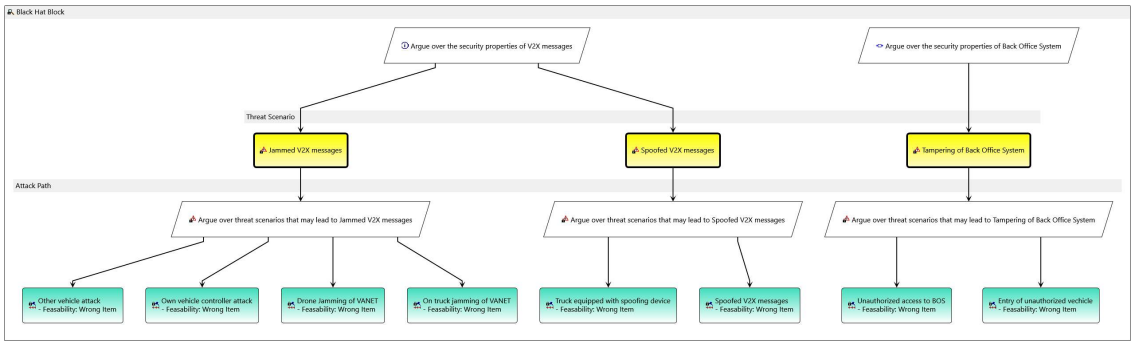
Incoming Security Goal	ISC Description
Avoid false vehicle security credentials	Vehicle SCMS information in Back Office Systems shall always be correct. (Prevention, Detection)
Avoid malicious (dangerous)	Erroneous or missing data on dynamic state of other vehicles must not occur.

Incoming Security Goal	ISC Description
erroneous beacons	(this can be done by prevention or detection of such cases, or mitigation of the consequence within designed safety margins) Note that this cybersecurity goal should be fulfilled by the other vehicles, although this cannot be trusted. Note also that although the communication of VANET can be secured, a deep attack in the C-ACC control system cannot be detected by VANET mechanisms.

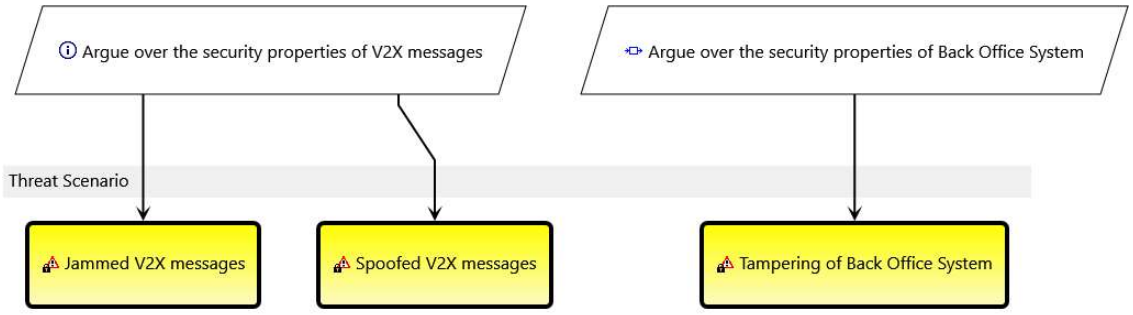
Black hat

Here we aim to identify the scenarios that might lead to not fulfilling the identified security goals, as well as the attack paths. To identify the threat scenarios, the threat model STRIDE can be used. Each threat scenario might be associated with multiple attack paths. For both threat scenarios and attack paths, we claim the negation.

Black Hat Diagram

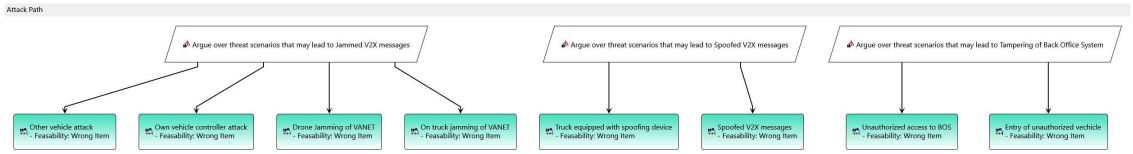


Threat Scenario



Threat	Impact Level	Threat Description	Strategy	Previous Node
Jammed V2X messages	Major		Argue over the security properties of V2X messages	V2X messages (Assets)
Spoofed V2X messages	Moderate			
Tampering of Back Office System	Major		Argue over the security properties of Back Office System	Back Office System (Assets)

Attack Paths



Attack Path	Feasability	Attack Path Description	Strategy	Previous Node
Entry of unauthorized vechicle	Low		Argue over the threat scenarios that may lead to Tampering of Back Office System	Tampering of Back Office System (ThreatScenario)
Unauthorized access to BOS	Medium			
VANET / C-ACC attack	Medium		Argue over the threat scenarios that may lead to Jammed V2X messages	Jammed V2X messages (ThreatScenario)
Spoofed V2X messages	Medium	Spoofed messages sent on the VANET Assuming new spoofing method.	Argue over the threat scenarios that may lead to Spoofed V2X messages	Spoofed V2X messages (ThreatScenario)
Truck equipped with spoofing device	Medium	Truck equipped with spoofing device, capable of intervening with the trucks own VANET messages		

Resolver

This block links the claims derived from the attack paths to the evidence. We assess the risk of the identified attack paths and based on the risk level, we choose how to treat the risk. Furthermore, requirements of risk treatments from the risk assessment level are expressed as claims.

Resolver Diagram

Risk Assessment

Risk Assessment Table:

Risk Assessment	Risk Level	Risk Description	Strategy	Previous Node
Back Office System tampering	2		Argue over the attack paths related to Entry of unauthorized vechicle	Entry of unauthorized vechicle (AttackPath)
			Argue over the attack paths related to Unauthorized access to BOS	Unauthorized access to BOS (AttackPath)
V2X jamming attack	3		Argue over the attack paths related to VANET / C-ACC attack	VANET / C-ACC attack (AttackPath)
V2X spoofing	2		Argue over the attack paths related to Spoofed V2X messages	Spoofed V2X messages (AttackPath)
			Argue over the attack paths related to Truck equipped with spoofing device	Truck equipped with spoofing device (AttackPath)

Requirements & Security Goals

Requirement/Security Goal	R/S Goal Description	Previous Node
test		Back Office System tampering (RiskAssessment)
Multi factor authentication	Access to the BOS shall require multi factor authentication.	
Add safety margins to malicious trucks	The safety margins shall be increased for malicious C-ACC truck, during revocation, including also failing revocations due to non cooperation, until conforming to ACC parameters.	V2X jamming attack (RiskAssessment)
Apply effective beacon error detection	The beacon data of a cooperative vehicle shall be validated against ego sensor data and model reference data.	
Driver revokation alert	When a revokation is initiated the driver should immediately be informed using appropriate sound and vision aids.	
Exclude/revoke malicious truck	C-ACC trucks that are deemed to be malicious shall be revoked from the C-ACC platoon.	
Controller data signature check	All internal controller data shall have an end-to-end signature that shall be validated, at least, before control output.	
Secure VANET communication	The V2X communication shall utilize network level security protocol to allow only authentic messages, i.e. sent from authentic senders.	V2X spoofing (RiskAssessment)

4 Conclusion