Security Assurance Cases – State of the Art of an Emerging Approach

Citation for the original published paper (version of record):
Mohamad, M., Steghöfer, J., Scandariato, R. (2021). Security Assurance Cases – State of the Art of an Emerging Approach. Empirical Software Engineering, 26(4).
http://dx.doi.org/10.1007/s10664-021-09971-7

Empirical Software Engineering (2021) 26: 70
https://doi.org/10.1007/s10664-021-09971-7

Security assurance cases—state of the art of an emerging approach

Mazen Mohamad[1] · Jan-Philipp Steghöfer[1] · Riccardo Scandariato[2]

Accepted: 14 April 2021 / Published online: 18 May 2021

Abstract

Security Assurance Cases (SAC) are a form of structured argumentation used to reason about the security properties of a system. After the successful adoption of assurance cases for safety, SAC are getting significant traction in recent years, especially in safety-critical industries (e.g., automotive), where there is an increasing pressure to be compliant with several security standards and regulations. Accordingly, research in the field of SAC has flourished in the past decade, with different approaches being investigated. In an effort to systematize this active field of research, we conducted a systematic literature review (SLR) of the existing academic studies on SAC. Our review resulted in an in-depth analysis and comparison of 51 papers. Our results indicate that, while there are numerous papers discussing the importance of SAC and their usage scenarios, the literature is still immature with respect to concrete support for practitioners on how to build and maintain a SAC. More importantly, even though some methodologies are available, their validation and tool support is still lacking.

Keywords Security · Assurance cases · Systematic literature review

1 Introduction

A security assurance case (a.k.a. security case, or SAC) is a structured set of arguments that are supported by material evidence and can be used to reason about the security posture of a software system. SACs represent an emerging trend in the secure development of critical systems, especially in domains like automotive and healthcare. The adoption of security cases in these industries is compelled by the recent introduction of standards and legislation. For instance, the upcoming standard ISO/SAE 21434 on “Road Vehicles—Cybersecurity Engineering” includes the explicit requirement to create ‘cybersecurity cases’ to show that a vehicle’s computing infrastructure is secure.

Communicated by: Eric Bodden

✉ Mazen Mohamad
mazen.mohamad@gu.se

Extended author information available on the last page of the article.

The creation of a security case, however, is far from trivial, especially for large organizations with complex product development structures. For instance, some technical choices about the security case might require a change of the development process. The security case shown in Fig. 1 (and discussed in Section 2), e.g., requires that a thorough threat analysis is conducted throughout the product structure and at different stages of the development. If this analysis is not yet created during development, either a thorough re-organization of the way of working is necessary or the security case should have been structured in a different way. Also, the construction of a security case often requires the collaboration of several stakeholders in the organization, e.g., to ensure that all the necessary evidence is collected from the software and process artifacts.

Companies are thus facing the conundrum of making both urgent and challenging decisions concerning the adoption of SACs. In order to facilitate such an endeavor, this paper presents a systematic literature review (SLR) of research papers on security cases. It summarises academic research which has published a relatively large number of papers on the topic in recent years and therefore provides practitioners an overview of the state of the art. To the best of our knowledge, this is the first study of this kind in this field. This SLR collects most relevant resources (51 papers) and presents their analysis according to a rich set of attributes like, the types of argumentation structures that are proposed in the literature (threat identification—used in Fig. 1—being one option), the maturity of the existing approaches, the ease of adoption, the availability of tool support, and so on.

Ultimately, this paper presents a reading guide geared towards practitioners. To this aim, we have created a workflow describing the suggested activities that are involved in the adoption of security cases. Each stage of the workflow is annotated with a suggested reading list, which refers to the papers included in this SLR. We remark that the SLR also represents a useful tool for academics to identify research gaps and opportunities, which are discussed in this paper as well.

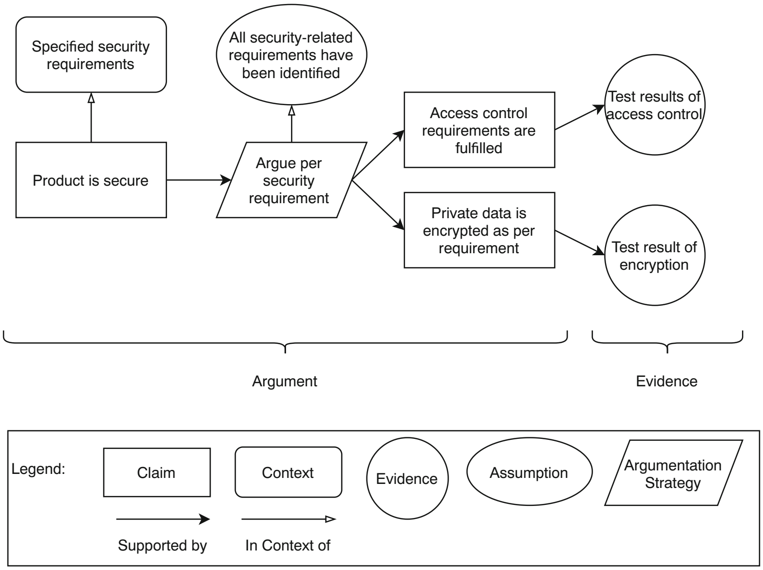
Specified security requirements
All security-related requirements have been identified
Product is secure
Argue per security requirement
Access control requirements are fulfilled
Private data is encrypted as per requirement
Test results of access control
Test result of encryption
Argument
Evidence
Legend:
Claim
Context
Evidence
Assumption
Argumentation Strategy
Supported by
In Context of

Fig. 1 An example of a SAC

The rest of the paper is structured as follows. In Section 2 we provide some background on assurance cases and discuss the related work. In Section 3 we describe the research questions and the methodology of this study. In Section 4 we list the papers included in this study and present the results of the analysis. In Section 5 we present a workflow for SAC creation and a reading guide for practitioners who want to adopt them. In Section 6 we further discuss the results and the lessons learnt from them. In Section 7 we discuss the threats to validity of this study. Finally, Section 8 presents the concluding remarks.

2 Background and Related Work

In this section, we first present background information about SACs, their main elements and their application areas as well as a simple example of a SAC. Afterwards, we discuss the related work.

2.1 Assurance Cases

Assurance cases are defined by the GSN standard (Group 2011) as *"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 can be documented in either textual or graphical forms. Figure 1 depicts a very simple example of an assurance case and its two main parts, i.e., the argument and the evidence. The case in the figure follows the GSN notation (Spriggs 2012), and consists of the following nodes: claim (also called goal), context, strategy, assumption (also called justification), and evidence (also called solution). At the top of the case, there is usually a high-level *claim*, which is broken down to sub-claims based on certain strategies. The claims specify the goals we want to assure in the case, e.g., that a certain system is secure. An example of a *strategy* is to break down a claim based on different security attributes. Claims are broken down iteratively until we reach a point where *evidence* can be assigned to justify the claims/sub-claims. Examples of evidence are test results, monitoring reports, and code review reports. The *assumptions* made while applying the strategies, e.g., that all relevant threats have been identified, are made explicit using the assumption nodes. Finally, the *context* of the claims is also explicitly defined in the context nodes. An example of a context is the definition of an acceptably secure system.

Assurance cases have been widely used for safety-critical systems in multiple domains (Bloomfield and Bishop 2010). An example is the automotive industry, where safety cases have been used for demonstrating compliance with the functional safety standard ISO 26262 (Palin et al. 2011; Birch et al. 2013; International Organization for Standardization 2011). However, there is an increasing interest in using these cases for security as well. For instance, the upcoming automotive standard ISO 21434 (International Organization for Standardization and Society of Automotive Engineers 2018) explicitly requires the creation of cyber-security arguments. SACs are a special type of assurance cases where the claims are about the security of the system in question, and the body of evidence justifies the security claims.

2.2 Related Work

To the best of our knowledge this study is the first systematic literature review on SACs. However, there have been studies covering the literature on safety assurance cases.

Nair et al. (2013) conducted a systematic literature review to classify artefacts which can be considered as safety evidence. The researchers contributed with a taxonomy of the evidence, and listed the most frequent evidence types referred to in literature. The results of the study show that the structure of safety evidence is mostly induced by the argumentation and that the assessment of the evidence is done in a qualitative manner in the majority of cases in contrast to quantitative assessment. Finally, the researchers list eight challenges related to safety evidence. The creation of safety cases was the second most mentioned one in literature according to the study. In our study, we focus on security rather than safety cases. We also review approaches for creating complete assurance cases, meaning that we look into both the argumentation and the evidence parts, in contrast to the study of Nair et al. (2013) which focuses on the evidence part only.

Maksimov et al. (2018) contributed with a systematic literature review of assurance case tools, and an extended study which focuses on assurance case assessment techniques (Maksimov et al. 2019). The researchers list 37 tools that have been developed in the past two decades and an analysis of their functionalities. The study also includes an evaluation of the reported tools on multiple aspects, such as creation support, maintenance, assessment, and reporting. In our study, we also review supporting tools for the creation of assurance cases, but we focus on the reported tools specifically for SAC.

Gade and Deshpande (2015) conducted a literature review of assurance-driven software design. The researchers provide a review of 15 research papers with an explanation of the techniques and methodologies each of these papers provide with regards to assurance-driven software design. This work intersects with our work in that assurance-driven software design can be used as a methodology or approach for creating assurance cases. However, unlike Gade et al. our study focuses on SAC, and is done in a systematic way.

Ankrum and Kromholz (2005) created a non-deterministic workflow for developing a structured assurance case. However, the proposed flow does not include anything related to tools or patterns usage. It does not consider the preliminary stage of considering a SAC either. Cyra and Gorski (2007) present the life-cycle, derivation procedure, and application process for a trust case template. All these artifacts, however, build on the argumentation strategy being derived from a standard, which is not always the case.

3 Research Method

We conducted a systematic literature review following the guidelines introduced by Kitchenham et al. (2007).

3.1 Research Questions and Assessment Criteria

This study aims at answering the following four research questions.

[RQ1] RATIONALE—In the literature, what rationale is provided to support the adoption of SAC?

In particular, we are interested in whether there are statements that go beyond the intuitive rationale of using SAC "for security assurance". For instance, our initial research (Mohamad et al. 2020) indicated that compliance with security standards and regulations is also an important driver. As shown in Table 1, to answer this research question we analyze the surveyed papers and extract two characteristics:

Table 1 Assessment criteria for RQ1 (rationale)

RQ1 criteria	Value
Motivation	E.g., compliance to standards, ensuring the fulfillment of security requirements, documenting security claims, …
Usage scenario	E.g., support for court case, assess security level of product or service, obtain certification, …

- *Motivation*, i.e., the reason for using SACs as stated by the researchers. We used two criteria for determining whether a certain study provides a motivation for using SAC. That is, the wording has to be explicit (i.e., there must be a reference to usage or advantage) and specific (i.e., providing some details).
- *Usage scenario*, i.e., scenarios in which SAC could be used to achieve additional goals, next to security assurance. We used the same criteria (explicit and specific mention) used for the motivation.

[RQ2] CONSTRUCTION—In the literature, which approaches are reported for the construction of SACs and which aspects do the approaches cover?

This question aims at inventorying the existing approaches for creating SAC, which is a challenging task for adopters. As shown in Table 2, we also assess the *coverage* of the approach, i.e., whether it can be used for creating the argumentation, for collecting the evidence, or both. Finally, for each covered part of the SAC, we summarise the approach with respect to the suggested *argumentation* strategy and the types of *evidence* to be used in creating SACs.

[RQ3] SUPPORT—In the literature, what practical support is offered to facilitate the adoption of SAC?

The purpose of this question is to understand the practicalities of creating and working with SAC. With reference to Table 3, first we study the approaches and identify the conditions (i.e., *prerequisites*) that have to be met in order for the outcome of the paper to be applicable. Second, we check whether the papers propose libraries of *patterns* or templatized SAC, as these are extremely useful for non-expert adopters. Third, we analyze the *tool support*. We check whether the paper suggests the usage of a tool for any of the activities related to SAC. In case it does, we extract the description of that tool, and whether it was created by the researchers or if it is a third party tool used in the paper. The last characteristic in this research question is the *notation* used to represent the SAC. The most common

Table 2 Assessment criteria for RQ2 (construction)

RQ2 criteria	Values
Coverage	Argumentation, Evidence, Generic (i.e., both)
Argumentation (if covered)	E.g., based on threat avoidance, …
Evidence (if covered)	E.g., collect test results, …

Table 3 Assessment criteria for RQ3 (support)

RQ3 criteria		Values
Prerequisites		E.g., threat modeling is performed, . . .
Patterns		E.g., a catalog or argumentation patterns . . .
Tool support	Tool mentioned	Yes / No
	Type of tool	Created / Used
Notation		Graphical (GSN, CAE), Textual, . . .

ones are GSN (Spriggs 2012), and CAE (Adelard 1998), but there are other notations such as plain text.

[RQ4] VALIDATION—In the literature, what evidence is provided concerning the validity of the reported approaches?

Our interest is to understand how the approaches and usage scenarios of SAC are validated (or supported by evidence). With reference to Table 4, we aim at identifying:

- The *type* of validation conducted in the study, e.g., case study, or experiment. Note that 'case study' is a widely used term to refer to worked examples (Easterbrook et al. 2008; Runeson and Höst 2009). In this work, we consider a validation conducted in an industrial context to be a case study (Yin and et al 2003), and those done within a research context to be illustrations. Experiments are studies in which independent variables are manipulated to test their effect on dependent variables (Easterbrook et al. 2008).
- The *domain* (i.e., application area) in which the validation is conducted.
- The *source* of the data used for the validation, e.g., a research project or a commercial product.
- Whether or not a SAC is *created* as part of the validation process.
- In case a SAC is created, we look for its *creators*. This characteristic has three possible values: academic authors, authors with industrial background, or third-party experts.
- The *validators*, i.e., the parties that conducted the validation with values the same as for creators.

Table 4 Assessment criteria for RQ4 (validation)

RQ4 criteria	Values
Type	Illustration, Case study, Experiment, Other
Domain	Medical, Automotive, Software engineering, . . .
Data source	Research project, Commercial product, . . .
SAC created	Yes / No
Creators	Academic authors, Industrial authors, third-party experts
Validators	Academic authors, Industrial authors, third-party experts

3.2 Performing the Systematic Review

We performed a search for papers related to SAC by means of 3 scientific search engines: IEEE Xplore, ACM Digital Library, and Elsevier Scopus. We selected these libraries, and did not include Google Scholar as, in our own prior experience which was confirmed by a preliminary search, the results from this search engine overlap with the results of the engines we mentioned.

3.2.1 Constructing the Search String

To maximize the chance of obtaining all relevant papers in the field, the search string used in the search engines must contain keywords that are commonly used in said papers. Therefore, prior to constructing the search string, we familiarized ourselves with the specific terminology used by researchers in the field of SAC. To do so, we conducted a manual search for papers related to SAC that were published in the past five years in the following venues: SAFECOMP, CCS, SecDev, ESSOS, ISSRE, ARES, S&P, Asia CCS, and ESORICS. The selection of the venues was based on their high visibility in the security domain.

Next, we created the search string for the selected libraries to identify papers that are potentially relevant for this study. In particular, we used two groups of keywords. The first group (line 1 below) is meant to scope the area of the study, while the second group (lines 2–4) included the terms referring to the parts of an assurance case. As a result, we formed the search string as follows:

```
1 (security OR privacy OR trust) AND
2 (claim OR argument OR evidence
3 OR justification OR’ assurance case’
4 OR assurance)
```

As a quality check for our search string, we ensured that we would find three relevant, known studies (Finnegan and McCaffery 2014a; Ben Othmane et al. 2014; Xu et al. 2017) with the search string. This was to make sure that our search string would return all three relevant studies, hence confirming its validity. We ran the query in IEEE Xplore and confirmed that the papers were returned.

3.2.2 Inclusion and Exclusion Criteria

The inclusion and exclusion criteria are shown in Table 5. This list has been created and fine-tuned by means of a calibration exercise involving two authors. We have invested significant time in performing an initial search of papers (prior to the systematic search) and discussing what papers should be included / excluded and why. This calibration made the application of this criteria straightforward later on, when filtering the results of the systematic search (as discussed below). The inclusion criteria are rather straightforward, considering the nature of this SLR. Concerning the exclusion criteria, we have decided to only consider studies written in English language, as this is the common language among the authors of this SLR. Further, SAC have been the focus of research only in recent times (although assurance cases, in general, have been around for much longer) and the field is rapidly evolving. Hence, we restricted our SLR to the past 15 years to avoid outdated results. We also excluded short papers, as answering our research questions requires studies with results rather than only

Table 5 Inclusion and exclusion criteria

Inclusion criteria
1. Studies addressing the creation, management, or application of SAC.
2. Studies related to security/privacy/trust assurance.
3. Studies related to security/privacy/trust argumentation.
Exclusion criteria
1. Studies written in any language other than English.
2. Studies published before 2004.
3. Short papers (less than 3 pages).
4. Studies focusing on risk/threat/hazard detection.
5. Studies addressing risk/threat/hazard analysis.
6. Studies addressing cryptography.
7. Studies focusing on security assessment/evaluation.
8. Studies about (only) safety assurance.

ideas. Finally, exclusion criteria 4–8 exclude studies that focus on topics that are marginally related to SAC but would not help us answer our research questions.

3.2.3 Searching and Filtering the Results

We executed the query on three libraries (IEEE Xplore, ACM Digital Library, and Scopus) in January 2019, and got the results shown in Table 6. In the case of Scopus, we limited the search to the domains of computer science and engineering. Also, because of the high number of returned results from Scopus, we decided to limit the included studies to the first 2000 after ordering the results based on relevance. We believe that the considered studies were sufficient, as the last 200 papers of the retained set from Scopus (i.e., papers 1801–2000) were all excluded when we applied the first filtering round (see below).

Afterthe systematic search had been applied, one author, who has been working in industrial projects about SAC with multiple partners in multiple domains (automotive and medical), performed an initial filtering (based on the inclusion and exclusion criteria) and tracked their confidence (high, medium and low) with each included / excluded paper. For the cases of medium to low confidence we held a series of meetings after each filtration round, where the three authors jointly discussed whether such papers should be included / excluded.

In the first filtering round, we applied the inclusion and exclusion criteria to the titles and keywords of all results (8440 papers). As shown in Table 6 this round reduced the number of

Table 6 Number of included studies after each round of filtration

Library	Papers	After filtering round		
		1st	2nd	3rd
IEEE Xplore	4513	118	23	22
ACM DL	1927	35	3	3
Scopus	2000	68	23	19
				+7 (snowballing)
Total	8440			51

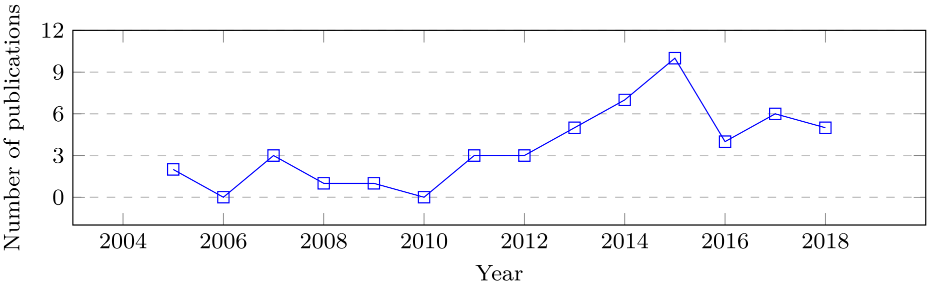

Fig. 2 Publication year of the included studies

studies to 211 papers. In the second filtering round, we applied the inclusion and exclusion criteria[1] to the abstracts and conclusions of the 211 remaining studies. After this step, the number of studies was reduced to 49. In the last filtering round, we fully read the remaining 49 papers, applied the inclusion and exclusion criteria on the whole text, and ended up with 43 included studies.

We also looked at the references mentioned by the included papers and performed *backward snowballing* (Wohlin 2014). In this step, we did not restrict the search to only peer-reviewed studies in order to allow for potential gray literature to be included. This resulted in additional 7 papers (including 2 technical reports) being included in our review. We looked into the references of these 7 papers, but this did not result in the inclusion of additional papers and we terminated the snowballing.

Finally, the authors kept monitoring the literature on the topic of SAC after the search was performed. This led to the inclusion of one additional paper, which is accessible through Scopus. In total, we thus included **51 studies**.

3.3 Analysis of the Included Papers

Once the final list of included studies was ready, we started the analysis phase. This was done in an iterative manner, where one author would use the infrastructure provided in Tables 1, 2, and 4 to prepare the analysis of a batch of papers (approximately 10 at a time). The outcome is then discussed in a group of the three authors as a means of quality control and calibration for the next batch.

4 Results

In this section, we provide a descriptive analysis of the included papers in this SLR, and then present the results and answers to our four research questions.

4.1 Descriptive Statistics

Figure 2 shows the years when our 51 included studies were published. The graph shows a peak of 10 publications in 2015, which indicates an increase in interest in the research field compared to previous years, especially the time between 2005 and 2012 where the number

[1] Except for exclusion criteria 1,2, and 8, which only needed to be applied once.

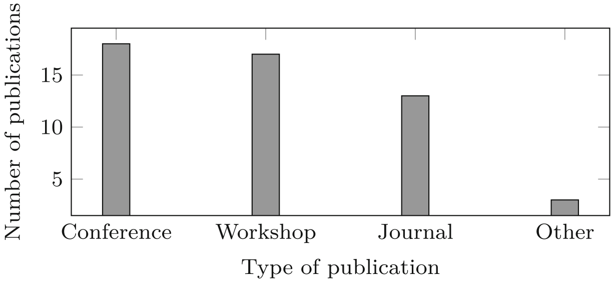

Fig. 3 Types of publication of the included studies

of publications was three or less each year. We decided to exclude the studies from 2019 in Fig. 2, as our search was conducted in that year and thus, results would necessarily be only partial. Including the results from that year would thys give a false indication of the trend compared to previous years.

Figure 3 shows the venues where the included studies were published. The graph shows that most of the publications were in conferences and workshops (18 and 17 respectively). 13 of the papers were published in journals, and three were technical reports.

We also looked into the authors of the selected papers to find the portion of the papers with at least one author from industry. We found that less than 25% (12 papers) (Cockram and Lautieri 2007; Goodger et al. 2012; Netkachova et al. 2015; Netkachova and Bloomfield 2016; Xu et al. 2017; Gacek et al. 2014; Rodes et al. 2014; Bloomfield et al. 2017; Netkachova et al. 2014; Gallo and Dahab 2015; Cheah et al. 2018; Ionita et al. 2017) included at least one author from industry.

To get an overview of the quality of the papers, we looked at the ranking of the venues for both conference and journal publications. We used CORE (2018), which has search portals for conferences and journals. The site gives the following ranking categories: A*—flagship venue in the discipline, A—Excellent venue, B—Good venue, and C— Other ranked venue. The ranking is based on the ERA ranking process (Australian Research Council 2018). For journals that were not ranked in Core (8 studies), we compared their impact factors to similar journals listed in CORE and assigned a ranking accordingly. Figure 4 shows the rankings of the venues that could be found in the portal's database. The column NA refers to conferences that were not found in the database.

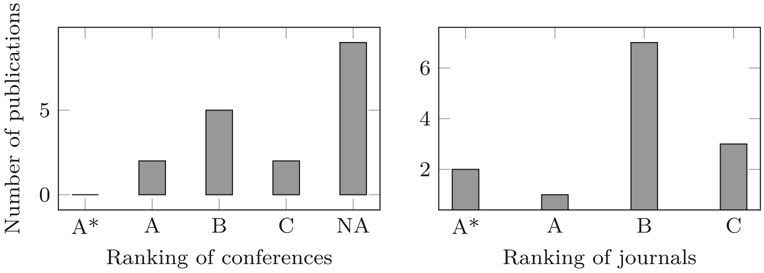

Fig. 4 Ranking of the venues of included journal papers according to the Core ranking portal

4.2 RQ1: Motivation

In order to find the rational reported in literature for the adoption of SAC, we looked into motivations and usage scenarios, as explained in Section 3.1. Some of the identified motivations in RQ1 could also be seen as usage scenarios. For example, *compliance with standards and regulation* could be seen as a motivation for using SAC, but also as a purpose for which SAC could be used.

4.2.1 Motivation

In the literature, papers often refer to the use of SAC as a means to build security assurance, which is a generic (and rather obvious) motivation. Instead, we looked for more specific motivations. In some of the papers, the motivation was made explicit in a separate section, or as the focus of the whole study (e.g. Knight 2015; Alexander et al. 2011). However, in most papers, this was briefly discussed either in the introduction and background sections, or as a part of motivating the used or suggested approach for creating SAC. If a study discusses only the generic SAC benefits, or is not being specific about the motivation (e.g., states that SAC provide security assurance in general), then we have categorised this paper as one that does not discuss any motivations for using SAC.

Table 7 shows all motivations found in our 51 sources. The results show that about 73% of the studies included at least one motivation for using SAC.

Categorizing the motivations resulted in the following categories:

- External forces: Compliance with standards and regulation (9 mentions), and compliance with requirements in case of suppliers (4 mentions).
- Process improvement: SAC helps in integrating security assurance with the development process (6 mentions). Moreover, they help factoring work per work items, and analyzing complex systems (2 mentions).
- Structure and documentation: The structure of SAC implies a way of work that reduces technical risks, and enhances security communication among stakeholders (7 mentions).
- Security assessment: SAC help in assessing security and spotting weaknesses in security for the systems in question (6 mentions). Hence, they help building confidence in the those systems (3 mentions).
- Knowledge transfer: It is a proven approach in safety which has been used effectively for a long time, and could be similarly in security (5 mentions).

4.2.2 Usage Scenarios

While SACs are usually used to establish evidence-based security assurance for a given system, researchers have reported cases where SAC could be used to achieve different goals. We looked into studies that focus on using SAC for a purpose other than security assurance, or for a purpose that is specific to a certain domain (e.g., security assurance for medical devices) or context (e.g., security assurance within the agile framework).

Table 8 shows the usage scenarios of SAC found in literature. We were able to extract usage scenarios from 14 different papers (28% of the total number of papers). The usage scenarios we found show a wide range of applications of SAC. Seven of the papers suggest using SAC for *evaluating different parts of the system or its surroundings*. For five papers, the use of SAC can be categorised as *providing process and life-cycle support*. One paper

Table 7 RQ1—Papers stating the motivations for using SAC

Study	Motivation
External forces	
Ankrum and Kromholz (2005)	Comply with standards and regulation
Calinescu et al. (2017)	Comply with security requirements of safety-critical systems
Cyra and Gorski (2007)	Comply with standards and regulation
Finnegan and McCaffery (2014a)	Comply with regulation and maintain confidence in the product in question
Finnegan and McCaffery (2014b) (2)	Comply with regulation
He and Johnson (2012)	Reason about cybersecurity policies and procedures
Mohammadi et al. (2018)	Learn from the safety domain where it is a proven approach
Ray and Cleaveland (2015)	Comply with regulation and internal needs from cyber-physical systems' manufacturers
Sklyar et al. (2017a, b, 2019) (2)	Comply with standards
Sljivo and Gallina (2016)	Comply with standards and regulation
Strielkina et al. (2018)	Comply with security regulation
Knowledge transfer	
Goodger et al. (2012)	Learn from the safety domain to integrate oversight for safety and security
Ionita et al. (2017)	Learn from the safety domain where it is a proven approach
Netkachova et al. (2014) (2)	Learn from the safety domain where it is a proven approach
Poreddy and Corns (2011)	Learn from the safety domain, where it is a proven approach
Sklyar and Kharchenko (2016)	Learn from the safety domain, where it is a proven in-use approach
Process improvement	
Ben Othmane and Ali (2016)	Trace security requirements and assure security during iterative development.
Ben Othmane et al. (2014)	Assure security during iterative development
Cheah et al. (2018)	Cope with the increasing connectivity of systems
Cockram and Lautieri (2007)	Reduces both technical and program risks through process improvement
Gallo and Dahab (2015)	Factor analytical and implementation work per component, requisite, technology, or life-cycle
Lipson and Weinstock (2008)	Help analyzing complex systems
Netkachova and Bloomfield (2016)	Tackle security issues which have intensified challenges of engineering safety-critical systems
Weinstock et al. (2007)	Include people and processes in security assurance in addition to technology

Table 7 (continued)

Study	Motivation
Security assessment	
Alexander et al. (2011)	Help security evaluators to focus their attention on critical parts of the system
Bloomfield et al. (2017)	Ensure the fulfillment of security requirements
Finnegan and McCaffery (2014b)	Improve overall security practices and demonstrate confidence in security
Hawkins et al. (2015)	Justify and assess confidence in critical properties
Knight (2015)	Spot security related weaknesses in the system
Poreddy and Corns (2011)	Assist in identifying security loopholes while changing the system
Rodes et al. (2014)	Measure software security
Strielkina et al. (2018)	Acquire an input for decision making of requirement conformity
Vivas et al. (2011)	Acquire confidence that the security of the system meets the requirements
Structure and documentation	
Agudo et al. (2009)	Incorporate certifications and evaluation methods in an evidence-based structure
Alexander et al. (2011)	Summarize security thinking when vendors are involved
Finnegan et al. (2013)	Communicate and report achieved security level
Knight (2015)	Document rational for security claims
Netkachova et al. (2015)	Aid in communication as it provides a summary of issues and their interrelationship
Patu and Yamamoto (2013b)	Aid in the survival of modern system, with respect to security challenges
Ray and Cleaveland (2015)	Comply with internal needs from cyber-physical systems' manufacturers

suggests to use SAC to communicate between organisations involved in developing and using medical devices and one paper uses SAC to teach students about information security.

4.3 RQ2: Approaches

We were able to find 26 different approaches in the literature. These studies focus on creating either one part of SACs (argumentation or evidence) or both parts. Table 9 shows these approaches, which part/s of SAC they cover, which argumentation strategies they use to divide the claims and create the arguments, and the evidence used to justify the claims in the approaches. We categorize the approaches as follows:

- *Integrating SAC in the development life-cycle*: These approaches suggest mapping the SAC creation activities to the development activities to integrate SACs in the development and security processes (Agudo et al. 2009; Ben Othmane et al. 2014; Ray and

Table 8 RQ1—Papers relevant to understanding the usage scenarios

Study	Usage scenario
Evaluating different parts of the system or its surroundings	
Graydon and Kelly (2013)	Evaluation of security standards, i.e., to make sure that conformance with a security standard is sufficient to achieve an acceptable and adequate level of security.
Haley et al. (2005)	Proving the achievement of security requirements satisfaction. The SAC arguments are constructed as formal claims about the system behaviour called outer arguments, and informal inner arguments, which include but are not limited to: sub-claims, supporting facts, reliability, and trustworthiness claims.
Masumoto et al. (2013)	Validation of service service grade achievement. The grade index value is used to quantify security operation claims, e.g., availability level, and the SAC is used to assure that the service meets that value.
Mohammadi et al. (2018)	Ensuring trustworthiness in cyber-physical systems using trustworthiness cases. These cases are incorporated with trustworthiness the development process to actively evaluate while developing the system
Netkachova et al. (2014) (2)	Evaluation of security of critical infrastructures. The scenario suggests using an inter-dependency analysis method to assign a quantitative evaluation of the reliability of the evidence of the SAC. This would be used to support decisions related to the CI system's security
Rodes et al. (2014)	Measuring software security based on confidence in security argument. The claims are annotated with a confidence level, and the SAC is supplemented with confidence arguments providing an assurance of the quality of the SAC's evidence and context items to make sure that these items properly support their associated claims
Yamamoto (2015)	Evaluation of system architecture based on security claims. It suggest assigning values to evidence ranging from −2 to 2 based on how satisfied the evidence is.
Providing process and life-cycle support	
Agudo et al. (2009)	Integrating security engineering and assurance based development using SAC to help addressing security in a systematic and comprehensive way throughout the life-cycle.
Ben Othmane and Ali (2016), Ben Othmane et al. (2014)	Controlling the impact of incremental development on security assurance using SAC. The SACs are developed along-side the security features and are used to actively ensure the completeness of security tests of these features.

Table 8 (continued)

Study	Usage scenario
Bloomfield et al. (2017)	Using SAC in the development of security strategies and policies. This is done by using the arguments of the SAC as a structuring mechanisms for the objectives of the intended policy/strategy.
Goodger et al. (2012)	Supporting the protection of critical information infrastructure assets by providing a method to manage the life-cycles of the assets using SAC. Each asset is represented by one SAC, and a larger SAC is used to group multiple individual ones. The suggested SACs are living documents which take a defined evidence body as as input, and have defined monitoring points.
Other uses	
Finnegan et al. (2013)	Reporting the achieved security level. The proposed SACs help to show the achievement of security capability and communicating them between Health Delivery Organizations (HDO) and Medical Device Manufacturers (MDM)
Gallo and Dahab (2015)	Using SAC to teach information security. The students are provided with a SAC and asked to learn from it and extract security requirements and goals in order to use them in their own projects.

Cleaveland 2015; Vivas et al. 2011), as well as assurance case driven design (Sklyar and Kharchenko 2016, 2017a, b, 2019). In general, these approaches suggest that the different stages of software development (requirements, design, implementation, and deployment) correspond to different abstraction levels of the security claims that can be made on the system. The hierarchical structure of SAC makes it possible to document these claims at every development stage as well as the dependencies to claims in the later or earlier stages (Vivas et al. 2011). This also applies to incremental development, e.g., using the SCRUM method (Ben Othmane et al. 2014). Updating SACs during the development life-cycle is, however, essential for these approaches to work. Hence, conducting these updates has to be included as a mandatory activity in the security life-cycle of the system under development (Sklyar and Kharchenko 2016).

– *Using different types of AC for security*: These approaches suggest using different types of assurance cases other than SAC for security assurance. These types are: *(i)* trust cases, which are based on assurance cases templates derived from the requirements of security standards (Cyra and Gorski 2007); *(ii)* trustworthiness cases, which focus mainly on addressing users' trust requirements (Górski et al. 2012; Mohammadi et al. 2018); and *(iii)* combined safety and security cases (Cockram and Lautieri 2007). This approach combines safety and security principles to create assurance cases with the main goal of achieving acceptable safety. The resulting cases have separate top claims for safety and security followed by separate argumentation; *(iv)* dynamic assurance cases (Calinescu et al. 2017), an approach for generating arguments and evidence based on run-time patterns for the assurance cases of self-adaptive systems; *(v)* multiple viewpoint assurance cases where security is treated as an assurance viewpoint (Sljivo and Gallina 2016). The approach suggests to reuse AC artefacts by building multiple-viewpoint AC using contracts, and introduces an algorithm for a model transformation from a contract meta model into an argumentation meta model; and *(vi)* dependability cases with focus on security (Patu and Yamamoto 2013a).

- Documenting and visualizing SAC: These studies give guidelines of how to document a SAC, and visualize it (Poreddy and Corns 2011; Coffey et al. 2014; Weinstock et al. 2007). In this category there are papers that focus on a specific part of SAC. These are:

  Argumentation-centric: These approaches focus on the argumentation part of the SACs. Different strategies are suggested in literature: security standards-based argument (Finnegan and McCaffery 2014a, b; Ankrum and Kromholz 2005), and satisfaction argument (Haley et al. 2005). Structures of argumentation found in literature are: model-based (Hawkins et al. 2015), and layered structure (Netkachova et al. 2015; Xu et al. 2017). Moreover, we have one study which suggests an automatic creation of argument graphs (Tippenhauer et al. 2014). As we can see, there is a variety of argumentation strategies used in these approaches, which shows that SAC arguments can be flexible and fit for most security artefacts present at organizations. However, this is not necessarily a positive characteristic when applied in industry, as it might result in heterogeneous SACs created in different parts of an organization. In consequence, it would be hard to apply quality metrics to the SACs and to combine SACs created for sub-systems. Hence, companies need to find a way to choose a suitable approach, but there is a lack of comparison of SAC creation approaches in literature, especially for different industries and in different contexts. This is further discussed in Section 6.2.

  Evidence-centric: These approaches focus mainly on different aspects of SACs' evidence. These aspects are: searching for evidence (Chindamaikul et al. 2014), collecting and generating evidence (Shortt and Weber 2015; Lipson and Weinstock 2008), and rating of potential artifacts to be used as evidence (Cheah et al. 2018). We conclude that even though the approaches cover main evidence-related activities, i.e., searching, locating, and rating, there are still essential parts missing, which are for example: assigning the evidence to claims, storing the evidence, and updating it over time. Similar to the argumentation-centric studies, the evidence-centric ones need to be more focused on the contexts in which they are applicable. Apart from the work of Cheah et al. (2018) which is done in the automotive domain, there is no focus towards domain specific SAC evidence work. We discuss this further in Section 6.5

4.3.1 Coverage

As shown in Table 9, 16 of the found approaches cover the creation of SACs including both argument and evidence, six focus on argument, and the remaining four on evidence.

Five out of the 16 studies to create argument and evidence of security cases did not include any examples of evidence to justify the claims.

In general, the level of detail in the studies varies significantly. For example in the studies which cover the creation of argument and evidence of SAC, we found papers providing a very high-level description of both how to create them and what to use them for (Ray and Cleaveland 2015; Poreddy and Corns 2011), while other papers had very detailed descriptions of how to extract the claims and divide them to create the arguments. However, the latter is often related to a specific context, e.g., self-adaptive systems (Calinescu et al. 2017). We also observed that these studies focus significantly more on the argument part than the evidence part. This is further discussed in Section 6.5.

Table 9 RQ2—Papers presenting approaches to construct SAC (A: Argument, E: Evidence, TR: Test Results, TVA: Threat and Vulnerability Analysis, CA: Code Analysis, BA: Bug Analysis, PA: Security Standards and Policies, RA: Risk Analysis, LA: Log Analysis, PD: Process Document, SA: Security Awareness and Training)

Approach	Coverage	Argumentation	Evidence
Integrating SAC in the development life-cycle			
Assurance-based development (Agudo et al. 2009)	A E	Requirements, system goals, system views and models	PA
Security assurance for incremental SD (Ben Othmane et al. 2014)	A E	Security goals	TR, CA
Integrating security engineering and AC development (Ray and Cleaveland 2015)	A E	Development life-cycle phases	–
Assurance Case Driven Design Sklyar and Kharchenko (2016, 2017a, b, 2019)	A E	Quality requirements, security properties, features, components, software layers, green IT principles	CA, TR
Security assurance driven SD (Vivas et al. 2011)	A E	Threats, vulnerabilities	–
Using different types of AC for security			
Dynamic assurance cases (Calinescu et al. 2017)	A E	Requirements	TR
TRUST-IT - trustworthiness arguments (Górski et al. 2012)	A E	Toulmin's argument (Toulmin 2003)	–
Trustworthiness cases (Mohammadi et al. 2018)	A E	Availability, threat analysis, goals satisfaction	RA, LA, DT, TVA
Evidence-based dependability case (Patu and Yamamoto 2013a)	A E	Vulnerabilities	TVA, SA, LA
Multiple-viewpoint AC (Sljivo and Gallina 2016)	A E	Contracts (pair of assumptions and guarantees)	TR
Trust-cases for security standards compliance (Cyra and Gorski 2007)	A	Risks	–
Dependability by contract (Cockram and Lautieri 2007)	A	Vulnerabilities, threats, and mitigation	TVA
Documenting and Visualizing SAC			
Mapping SAC to standards (Ankrum and Kromholz 2005)	A E	Security standard description	–

Table 9 (continued)

Approach	Coverage	Argumentation	Evidence
Concept map-based (Coffey et al. 2014)	A E	Vulnerabilities	–
Risk based approach Finnegan and McCaffery (2014a, b)	A E	Security capabilities, mitigation controls	TVA, LA
Layered Approach (Netkachova et al. 2015)	A E	Source of security requirements, changes during life-cycle	TVA, PA
Documenting AC for Security (Poreddy and Corns 2011)	A E	Security properties	TR
Arguing security (Weinstock et al. 2007)	A E	Prevention and detection	TR, TVA, SA
Satisfaction arguments (Haley et al. 2005)	A	Security requirements	–
Model-based assurance (Hawkins et al. 2015)	A	Software components	–
Automatic generation of argument graphs (Tippenhauer et al. 2014)	A	Security goals	–
Layered Argument strategy (Xu et al. 2017)	A	Assets, threats	–
Systematic Security Evaluation (Cheah et al. 2018)	E	–	TR
Document retrieval and concept analysis (Chindamaikul et al. 2014)	E	Security properties	TR, BA
Evidence-based security properties' assurance (Lipson and Weinstock 2008)	E	–	PD, TR, SA
Hermes Targeted fuzz testing (Shortt and Weber 2015)	E	–	TR

4.3.2 Argumentation

Argumentation is a very important part of SAC. The argumentation starts with a security claim, and continues as the claim is being broken down into sub-claims. The strategy is used to provide a means by which claims are broken down. Each level of the argumentation could be done with a specific strategy. Hence, one SAC might have one or more argumentation strategies as is the case in some of the included studies in this SLR, e.g., Agudo et al. (2009) and Mohammadi et al. (2018).

We looked for an explicit mention of the used strategy. If none was provided, we analysed the example cases to find the used argumentation strategy. Table 9 shows the approaches we found in literature with the respective argumentation strategies used in each of them.

When regarding argumentation strategies in the context of the different approaches, we could not find any correlation between the two. For instance, different approaches which integrate SAC within the development life-cycle use different argumentation strategies (e.g., requirements Agudo et al. 2009 and development phases Ray and Cleaveland 2015). The most common strategy depends on the output of a threat, vulnerability, asset or risk analysis (8 papers) (Cockram and Lautieri 2007; Coffey et al. 2014; Cyra and Gorski 2007; Mohammadi et al. 2018; Patu and Yamamoto 2013a; Vivas et al. 2011; Xu et al. 2017; Weinstock et al. 2007). Other popular strategies are breaking down the claims based on the requirements or more specifically quality requirements and even more specifically security requirements (5 papers) (Agudo et al. 2009; Calinescu et al. 2017; Haley et al. 2005; Netkachova et al. 2015; Sklyar and Kharchenko 2017b), and arguing based on security properties, e.g., confidentiality, integrity and availability (5 papers) (Chindamaikul et al. 2014; Finnegan and McCaffery 2014a; Mohammadi et al. 2018; Poreddy and Corns 2011; Sklyar and Kharchenko 2017b). Additionally, researchers also used system and security goals (4 papers) (Agudo et al. 2009; Ben Othmane et al. 2014; Mohammadi et al. 2018; Tippenhauer et al. 2014), software components or features (3 papers) (Agudo et al. 2009; Hawkins et al. 2015; Sklyar and Kharchenko 2017b), security standards and principles (2 papers) (Ankrum and Kromholz 2005; Sljivo and Gallina 2016), pre-defined argumentation model (1 paper) (Górski et al. 2012), and development life-cycle phases (1 paper) (Ray and Cleaveland 2015).

4.3.3 Evidence

Even though evidence is a very important and complex part of SAC, only four of 26 included approaches focused on it. Even in the approaches which cover argument and evidence of SACs, there was a much deeper focus on the argumentation than the evidence, which explains why five out of these did not even include an example of what evidence would look like. We found evidence either by looking for explicit mentions in the articles or by extracting the evidence part from the reported SACs. Table 9 shows the approaches we found in literature with the respective evidence types used in each of them.

The most common types of evidence reported in literature are *test results (TR)* (12 papers) (Ben Othmane and Ali 2016; Calinescu et al. 2017; Cheah et al. 2018; Chindamaikul et al. 2014; Lipson and Weinstock 2008; Poreddy and Corns 2011; Shortt and Weber 2015; Sklyar and Kharchenko 2016, 2017a, b, 2019; Sljivo and Gallina 2016) and different types of analysis. These analysis include threat and vulnerability (TVA) (Cockram and Lautieri 2007; Finnegan and McCaffery 2014a, b, Patu and Yamamoto 2013a), code (CA) and bug (BA) (Chindamaikul et al. 2014; Ben Othmane and Ali 2016; Sklyar and Kharchenko 2016, 2017a, b, 2019), security standards and policies (PA) (Agudo et al. 2009; Netkachova et al. 2015), risk (RA) (Mohammadi et al. 2018), and log analysis (LA) (Mohammadi et al. 2018; Patu and Yamamoto 2013a). Cheah et al. (2018) present a classification of security test results using security severity ratings. This classification can be included in the security evaluation, which may be used to improve the selection of evidence when creating SACs. Chindamaikul et al. (2014) investigate how information retrieval techniques, and formal concept analysis can be used to find security evidence in a document corpus. Shortt and

Weber (2015) present a method to apply fuzz testing to support the creation of evidence for SACs.

Other types of evidence reported in literature include *process documents (PD)* (Lipson and Weinstock 2008), *design techniques (DT)* (Mohammadi et al. 2018), and *security awareness and training (SA)* (Patu and Yamamoto 2013a; Lipson and Weinstock 2008; Weinstock et al. 2007). Lipson and Weinstock (2008) describe how to understand, gather, and generate multiple kinds of evidence that can contribute to building SAC.

4.4 RQ3: Support

In this section, we list our results from reviewing the practical support to facilitate the adoption of SAC reported in literature. Specifically, we report on the tools used to assist in any of the SAC activities, e.g., creation and maintenance, the prerequisites of the approaches, and patterns for creating SAC.

4.4.1 Tools

We found 16 software tools which have been used one way or another in the creation of SAC in literature. Seven of the found tools were created by researchers. Four of these seven target assurance cases in general (Fung et al. 2018; Gacek et al. 2014; Hawkins et al. 2015; Tippenhauer et al. 2014), while the remaining three are created to be used in the creation of SAC specifically (Ben Othmane and Ali 2016; Cheah et al. 2018; Shortt and Weber 2015). Table 10 shows the tools and the respective studies in which they are used. A brief description of the main functionalities of the tools, as well as whether the tools are created or used by the authors are also presented. There are four main types of reported tools. In the following, we list the tools of each type, and we discuss the main features of each tool as reported in the studies:

- Creation tools: used to create and document assurance cases in general.
- Argumentation tools: focus mainly on the creation of the argumentation part of SAC.
- Evidence tools: focus on the creation of SAC evidence.
- Support tools: several studies reported supporting tools to assist the creators of SAC in the analysis needed for creating them, e.g., by helping users determine the relevance of a given document to be used as evidence (Chindamaikul et al. 2014).

4.4.2 Prerequisites

Prerequisites are the conditions that need to be met before an approach presented in a study can be applied. We found prerequisites in the included studies by checking the inputs of the proposed outcomes (approaches, usage scenarios, tools, and patterns). If an input is not a part of the outcome itself, we considered it to be a prerequisite to that outcome. Table 11 shows the prerequisites we found along with the respective type of study for each. There are 17 reported prerequisites. The majority belong to approaches (11) (Chindamaikul et al. 2014; Cockram and Lautieri 2007; Cyra and Gorski 2007; Hawkins et al. 2015; Patu and Yamamoto 2013a; Ankrum and Kromholz 2005; Cheah et al. 2018; Sljivo and Gallina 2016; Tippenhauer et al. 2014; Vivas et al. 2011; Xu et al. 2017) while the remaining ones belong to usage scenarios (2) (Bloomfield et al. 2017; Goodger et al. 2012), patterns (2) (Patu and Yamamoto 2013b; He and Johnson 2012), and tools (1) (Gacek et al. 2014). We categorize prerequisites as follows:

Table 10 RQ3—Tools supporting the creation, documentation, and visualization of SAC (U: Used, C: Created)

Study	Tool support	Description	
Creation tools			
Poreddy and Corns (2011)	Adelard Safety Case Editor (ASCE) (Adelard 2003)	Supports the creation and visualisation of AC. It supports multiple notations, and enables assigning multiple formats of data in the bodies of AC nodes, e.g., text, tables, and images. Additionally, allows validation of the AC structure based on the rules of a notation or based on user-defined rules.	U
Ankrum and Kromholz (2005)	Adelard Safety Case Editor (ASCE) (Adelard 2003)		U
Finnegan et al. (2013)	TurboAC (GessNet 2011)	Enables converting artefacts of different formats used to create assurance cases, e.g., tabular or XML into HTML files which can be viewed and navigated with web browsers. Also allows electronically submitting assurance cases to external authorities.	U
Gacek et al. (2014)	Resolute	An open source software which enables to automatically construct assurance cases based on models which use the Architecture Analysis and Design Language (AADL).	C
Patu and Yamamoto (2013a)	D-Case Editor (Matsuno et al. 2010)	Used to document and visualize assurance cases. Includes a library of patterns which can assist the users in creating the cases.	U
Argumentation tools			
Hawkins et al. (2015)	Instantiation program (no specific name)	A model-based tool which takes GSN argument patterns and different information models as input. The information models hold relevant information for AC arguments, such as design models. The tool identifies the elements required to instantiate the GSN model, and outputs an instantiated model.	C

Table 10 (continued)

Study	Tool support	Description	
Ionita et al. (2017)	OpenArgue (Yu et al. 2011)	Provides editors and the ability to derive graphical arguments from textual requirements specifications. Users can specify inter-argument relationships, e.g., where one argument can mitigate another.	U
	ArgueSecure (Ionita et al. 2016)	Allows users to collaboratively work on argumentation spreadsheets, designed to decompose the arguments into claims, assumptions and facts.	U
Tippenhauer et al. (2014)	CyberSAGE (Singapore 2015)	Allows users to integrate information from different sources, e.g., network topology and attacker models, to automatically generate security arguments.	C
Calinescu et al. (2017)	UPPAAL (Behrmann et al. 2006)	A verification tool suite used to generate evidence to show the achievement of a claimed goal. Also verifies that a model satisfies pre-defined correctness properties.	U
Shortt and Weber (2015)	Hermes	Provides dynamic code coverage analysis which can be used as SAC evidence.	C
Cheah et al. (2018)	Software tool (no specific name)	Semi-automated tool for penetration testing with features such as: identifying open ports, spoofing a device and scanning of log files. The output of the tool is used to create the body of evidence to be used in a SAC.	C
Support tools			
Ben Othmane and Ali (2016)	Meld (Willadsen 2011)	Visualizes differences between different files and helps merging these files.	U
	SECUREAGILE	Traces the impact of code changes on security to support the iterative development of security features with help of SAC.	C
Chindamaikul et al. (2014)	Concept lattice	Helps users to determine the relevance of a given document.	U
Fung et al. (2018)	MMINT-A	Automated change impact assessment for SAC.	C

Table 10 (continued)

Study	Tool support	Description	
Górski et al. (2012)	NOR-STA (G.U. of Technology 2010)	A set of services used for editing and assessing argumentation of assurance cases. Also acsts as a repository to store evidence used in SAC.	U

- Usage of specific format (Gacek et al. 2014; Hawkins et al. 2015; Sljivo and Gallina 2016): In this category, studies require the use of artefacts which have specific formats to achieve the purpose of the study.
- Usage of specific documents and repositories (Chindamaikul et al. 2014; Cockram and Lautieri 2007; He and Johnson 2012; Patu and Yamamoto 2013a; Tippenhauer et al. 2014; Vivas et al. 2011): The studies in this category use specific repositories and documents for retrieving required data for building or using SAC.
- Usage of security standards (Ankrum and Kromholz 2005; Cyra and Gorski 2007): The studies in this category require the use of security standards to create SAC or make use of them.
- Existence of analysis and modelling (Cheah et al. 2018; Goodger et al. 2012; Patu and Yamamoto 2013b; Xu et al. 2017): The studies in this category require the existence or performing certain analysis and models to achieve their purpose.
- Existence of special expertise (Bloomfield et al. 2017): The one study in this category relies on expertise provided by an external safety regulator.

4.4.3 Patterns

Reoccurring claims and arguments in SAC can be subsumed in patters. They can save the creators of SACs a lot of time and effort. We found ten studies which deal with patterns. Six of these create their own argumentation patterns (Finnegan and McCaffery 2014a, b; He and Johnson 2012; Patu and Yamamoto 2013b; Poreddy and Corns 2011; Xu et al. 2017). The remaining four include usage of patterns (Hawkins et al. 2015; Tippenhauer et al. 2014), a guideline for creating and documenting security case patterns (Weinstock et al. 2007), and a catalogue of security and safety case patterns (Taguchi et al. 2014). Since we we only considered patterns created and used for SAC, we excluded those studies in which patterns are borrowed from the safety domain, e.g., Calinescu et al. (2017).

Table 12 shows the studies that deal with SAC patterns. While the created patterns cover an important aspect, namely abstraction, it is not clear how re-usable or generalize-able they are. Some patterns are derived from various security standards, e.g., Finnegan and McCaffery (2014a) and Taguchi et al. (2014) (these are usually from the medical domain where security standardization is more mature compared to other security-critical domains), and one from lessons learned from security incidents (He and Johnson 2012), but none is derived from previous applications of SAC in industry. Another observation we made is that the patterns focus heavily on the argumentation part of SAC in contrast to the evidence part. Only few studies provided examples of evidence that can be used in a given pattern (Poreddy and Corns 2011; Taguchi et al. 2014; Weinstock et al. 2007). However, these examples are specific to the context of the studies, and leaves the abstraction to the reader, with the notable exception of the examples provided by Weinstock et al. (2007).

Table 11 RQ3—Papers discussing the prerequisites of SAC approaches, usage scenarios, and tools

Study	Type	Prerequisites
Usage of specific format		
Sljivo and Gallina (2016)	Approach	Re-usability of assurance cases by using *safety contracts created using the Safety Element out-of-context Meta-model (SEooCMM).*
Hawkins et al. (2015)	Approach	Model-based approach for creating assurance cases based on GSN and an extended model of the structured assurance case meta-model by the OMG (2020). Requires reference information models (e.g., design and analysis models) and a weaving model (which connects the reference models to the GSN pattern) as inputs.
Gacek et al. (2014)	Tool	Proposes a tool for automatic generation of SAC which requires a system model specified in the Architecture Analysis and Design Language (AADL) (Feiler and Gluch 2012).
Usage of specific documents and repositories		
Chindamaikul et al. (2014)	Approach	Creation of SAC using information retrieval techniques based on an *existing repository of evidence.*
Patu and Yamamoto (2013a)	Approach	Creation of SAC based on a pre-defined list of . common risks, vulnerabilities and solutions in the domain
Cockram and Lautieri (2007)	Approach	Creation of SAC argument uses *Module boundary contracts* identified from the specification of the system in question.
Tippenhauer et al. (2014)	Approach	Automatic SAC argument generation requires the use of *extension templates*. These templates are formalization of sub-argument patterns.
Vivas et al. (2011)	Approach	The approach suggests integrating SAC within SDLC ((Software Development Life-Cycle)), which requires the existence of a *Well defined SDLC process*.
He and Johnson (2012)	Pattern	Patterns are created based on a *repository of lessons learned and recommendations* from previous security incidents.
Usage of security standards		
Cyra and Gorski (2007)	Approach	Cyra et al. Trust case templates are based on *security standards* and restructure the standards' information, e.g., their requirements.

Table 11 (continued)

Study	Type	Prerequisites
Ankrum and Kromholz (2005)	Approach	Applied assurance cases to the requirements of . three standards in order to study the applicability and problems of that approach. The authors used the learning outcome to create a practical SAC, but also used artefacts from one of the standards
Existence of analysis and modelling		
Cheah et al. (2018)	Approach	Construction and severity classification of SAC evidence based on a scripted attack tree and manual threat modelling. *Asset analysis and models* are required.
Existence of special expertise		
Xu et al. (2017)	Approach	Requires an *asset analysis and model* as a basis for a layered approach for creating SAC.
Bloomfield et al. (2017)	Usage scenario	Suggests using SAC for developing security strategy and policies. Most of the arguments and evidence are derived from the *expertise of a safety regulator*.
Goodger et al. (2012)	Usage Scenario	Requires an *asset analysis* to identify the assets for which the SAC will be created in order to protect critical infrastructure.
Patu and Yamamoto (2013b) (2)	Pattern	Describe an *asset analysis* as the basis for identifying security patterns at the requirements phase of the development life-cycle.

4.4.4 Notations

Out of 51 studies, 41 specify at least one notation to be used for expressing and documenting a SAC. Table 13 shows the number of studies that use each notation, and lists them. The most common notation is the Goal Structure Notation (GSN) (Spriggs 2012) which is suggested by 27 studies. Another popular notation is the Claim Argument Evidence (CAE) (Adelard 1998) notation which is suggested by nine studies. Other notations are: text (6 studies), concept maps (Coffey et al. 2014) (1), and Claim-Argument-Evidence Criteria (CAEC) (Netkachova and Bloomfield 2016; Netkachova et al. 2014, 2015) notation which is extension of the CAE notation (3 studies of the same authors).

4.5 RQ4: Validation

We consider validation to be the process to show that an approach or tool for creating SAC works in practice or that an SAC can actually be used for a suggested usage scenario. In case validation is performed in a selected study, we looked for the type of validation, the

Table 12 RQ3—Papers presenting patterns

Study	Description of the pattern-based approach
Creation of patterns	
Finnegan and McCaffery (2014a, b)	Creation of security capability argument pattern using a risk-based approach. Argues for each security capability defined in a technical report for risk management in medical devices.
He and Johnson (2012)	Creation of generic cases which use security arguments that are informed by security incidents in healthcare organizations.
Patu and Yamamoto (2013b)	Creation of security patterns during the requirement phase of system development. One suggested pattern argues over security attributes.
Poreddy and Corns (2011)	Creation of assurance case patterns. Suggested argumentation strategies are: integrity, availability, reliability, confidentiality and maintainability.
Xu et al. (2017)	Creation of different argument patterns to be used in different layers to form a layered argument structure.
Usage of patterns	
Hawkins et al. (2015)	Usage of argument patterns as input to the model-based approach for building assurance case arguments. A suggested pattern argues over individual software components.
Tippenhauer et al. (2014)	Usage of argument patterns to automatically generate argument graphs. The paper includes five different patterns categorized into the categories inter-type and intra-type.
Other	
Taguchi et al. (2014)	A catalogue of safety and security case patterns. The patterns are derived from process patterns through a literature survey.
Weinstock et al. (2007)	A guideline of how to create and use SAC patterns. An example pattern is also presented.

domain of application, the source of data, whether a SAC is created during the validation, the creators of the SACs, and who performed the validation.

Table 14 shows these different aspects for the 36 studies which include a validation of the outcome. The majority of the outcomes were validated using illustrative cases (21), 11 were validated using case studies, and the remaining four used experiments (3) and observation as a part of an Action Design Research (ADR) (Sein et al. 2011) study.

The data sources vary among the validations, as can be seen in Table 14. We categorize these sources into three main categories:

Table 13 Studies which use each notation

Notation	Number	Study
GSN	27	Alexander et al. (2011), Ankrum and Kromholz (2005), Ben Othmane and Ali (2016), Ben Othmane et al. (2014), Calinescu et al. (2017), Chindamaikul et al. (2014), Cockram and Lautieri (2007), Finnegan and McCaffery (2014a, b) Fung et al. (2018), Goodger et al. (2012), Graydon and Kelly (2013), Hawkins et al. (2015), He and Johnson (2012), Ionita et al. (2017), Masumoto et al. (2013), Mohammadi et al. (2018), Patu and Yamamoto (2013a, b) Poreddy and Corns (2011), Ray and Cleaveland (2015), Rodes et al. (2014), Sljivo and Gallina (2016), Taguchi et al. (2014), Weinstock et al. (2007), Xu et al. (2017),
CAE	9	Yamamoto (2015) Alexander et al. (2011), Ankrum and Kromholz (2005), Bloomfield et al. (2017), Finnegan et al. (2013), Goodger et al. (2012), Ionita et al. (2017), Netkachova et al. (2014, 2015), Netkachova and Bloomfield (2016)
Text	6	Cheah et al. (2018), Cyra and Gorski (2007), Gacek et al. (2014), Gallo and Dahab (2015), Górski et al. (2012), Ionita et al. (2017)
CAEC	3	Sklyar and Kharchenko (2016, 2017b, 2019)
Concept maps	1	Coffey et al. (2014)

- Research, open source, and in-house projects (20) (Ankrum and Kromholz 2005; Chindamaikul et al. 2014; Cockram and Lautieri 2007; Coffey et al. 2014; Gacek et al. 2014; Haley et al. 2005; Hawkins et al. 2015; Mohammadi et al. 2018; Netkachova et al. 2015; Patu and Yamamoto 2013a; Poreddy and Corns 2011; Ray and Cleaveland 2015; Rodes et al. 2014; Shortt and Weber 2015; Sklyar and Kharchenko 2019; Sljivo and Gallina 2016; Strielkina et al. 2018; Tippenhauer et al. 2014; Vivas et al. 2011; Gallo and Dahab 2015)
- Commercial products / systems (9) (Ben Othmane and Ali 2016; Ben Othmane et al. 2014; Calinescu et al. 2017; Cheah et al. 2018; Goodger et al. 2012; Górski et al. 2012; Masumoto et al. 2013; Xu et al. 2017; Netkachova et al. 2014)
- Standards, regulation, and technical reports (7) (Bloomfield et al. 2017; Cyra and Gorski 2007; Finnegan and McCaffery 2014b; Fung et al. 2018; Graydon and Kelly 2013; He and Johnson 2012; Sklyar and Kharchenko 2017b)

SACs were presented in 31 out of the 36 validations. Representing a complete SAC is mostly not possible even in small illustrative cases due to the amount of information required to build one. However, how much of an SAC is represented in the included validations varies to a large extent. Some validations present an example of a full branch of SAC, i.e., a claim all the way from top to evidence (e.g., He and Johnson 2012), while others present very brief examples of SACs (e.g., Gallo and Dahab 2015).

Table 14 also shows who created the SACs in each study. In only two cases, experts were used to create the SACs. In the majority of the studies (28), the authors created the SACs. However, eight of the studies included authors from industry. These are shown in Table 14 as “Authors*” in the Creators column.

Table 14 also shows the domains in which the validation was conducted. The most common domains are Software Engineering (7) and Medical (7).

The last column in Table 14 shows the persons which performed the validation in each study. Out of the 36 included validations, only five used third parties to validate the outcomes. These were industrial partners in two cases (Ben Othmane and Ali 2016; Cheah et al. 2018), an external regulator (Bloomfield et al. 2017), one security expert (Coffey et al. 2014), and a group of security experts (Finnegan and McCaffery 2014b). In the remaining 31 validations, the authors performed the validation. However, eight of the studies included authors from industry. These are shown in Table 14 as “Authors*” in the Validator column.

5 SAC Creation Workflow

Based on the results of this systematic literature review, we have found that the outcomes described in the literature fall into one or more parts of the workflow depicted in Fig. 5.

We also realised that there is agreement in the literature that SAC are to be created in a top-down manner. This means that one starts from a top-claim which represents a high-level security goal and work their way through strategies and sub-claims all the way to the evidence. We have not seen approaches that, e.g., start from the existing evidence of a certain system and constructs claims out of them in a bottom-up fashion. However, this agreement is not expressed in sufficient level of detail in any one paper yet.

Hence, we have synthesised the existing knowledge into a generic workflow for the construction of SAC. Even though the literature might have some gaps and fallacies, this workflow is useful as a contextual learning guide for the readers to familiarize themselves with the different aspects of SAC creation.

There are five main blocks in the workflow. We will list and describe them in the remainder of this subsection. Additionally, Table 15 lists recommended papers from our systematic literature review for practitioner wanting to adopt SAC as well as researchers wanting to conduct studies in a specific area of SAC. The recommended papers focus on aspects related to the individual blocks and together provide a thorough investigation of each.

Study and Understand SAC Building SACs is not trivial and requires significant effort. Hence, before going ahead and creating them, it is important to understand what they are and what they can be used for. This step includes studying the structure of SACs, their benefits, what needs to be in place to create them, and their potential usage scenarios, e.g., standards and regulation compliance. Block number 1 in Fig. 5 shows the corresponding entity in the workflow.

Argumentation This block includes selecting the top claim to achieve, and the strategy to decompose this claim into sub-claims. This is a very important step, as selecting an argumentation strategy decides to a big extent which activities are needed to complete the SAC. For example, if a strategy using decomposition based on vulnerabilities is adopted, a vulnerability analysis of the system in question has to be conducted. A sub-block of the argumentation is the usage of patterns. Patterns help the creators of SACs to save time and effort by using pre-defined and proven structures. The creators could, however, decide not

Table 14 RQ4—Papers presenting a form of validation

Study	Domain	Data source	SAC	Creators	Validators
Case study					
Ben Othmane and Ali (2016)	Software Engineering	E-Commerce product	✓	Authors	3rd party
Bloomfield et al. (2017)	Safety	Regulatory organization	✓	Authors*	3rd party
Calinescu et al. (2017)	Marine, Trading	Underwater Vehicle System, Trading System	✓	Authors	Authors
Cheah et al. (2018)	Automotive	Vehicle infotainment system, diagnostics tool	✓	Authers*	3rd party
Fung et al. (2018)	Automotive	Power sliding door—Case from ISO26262 standard	✓	Authors	Authors
Goodger et al. (2012)	Critical infrastructure	Critical information Infrastructure		NA	Authors*
Górski et al. (2012)	Medical	A software for patient monitoring	✓	Authors	Authors
Graydon and Kelly (2013)	Security	Security standards	✓	Authors	Authors
He and Johnson (2012)	Medical	Lessons learned from security s incidents, security standards, policies, and procedure	✓	Authors	Authors
Xu et al. (2017)	Software Engineering	IM server	✓	Authors*	Authors*
Illustrative case					
Ankrum and Kromholz (2005)	Security	Research security project		Authors	Authors
Ben Othmane et al. (2014) (2)	Telecom	Commercial project	✓	Authors	Authors
Cockram and Lautieri (2007)	SafSec	Command and control system for locating persons	✓	Authors*	Authors*

Table 14 (continued)

Study	Domain	Data source	SAC	Creators	Validators
Coffey et al. (2014)	Software Engineering	SOA composite application	✓	Expert group	3rd party
Cyra and Gorski (2007)	Security	Security standard BS 7799-2	✓	Authors	Authors
Gacek et al. (2014)	Embedded Systems	Research project for unmanned air vehicles	✓	Authors*	Authors*
Haley et al. (2005)	Software Engineering	Example HR system		NA	Authors
Hawkins et al. (2015)	Model-Based Engineering	Cryptographic controller system	✓	Authors	Authors
Mohammadi et al. (2018)	Medical	OPerational Trustworthiness Enabling Technologies (OPTET) research project	✓	Authors	Authors
Netkachova et al. (2015)	Aviation	A security gateway data-flow controller	✓	Authors*	Authors*
Patu and Yamamoto (2013a) (2)	Networking	Research e-learning project	✓	Authors	Authors
Poreddy and Corns (2011)	Aviation	Avionic mission control computer system	✓	Authors	Authors
Ray and Cleaveland (2015) n	Medical	A medical cyber-physical system for pumping insulin	✓	Authors	Authors
Rodes et al. (2014)	Security	Example scenario with confidence t properties measurement	✓	Authors*	Authors*
Shortt and Weber (2015)	Software Engineering	Java-based open source library (Crawler4J)		NA	Authors
Sklyar and Kharchenko (2017b) (3)	SafSec	Requirements derived from safety and security standard	✓	Authors	Authors
Sklyar and Kharchenko (2019) (4)	Medical	Example medical system	✓	Authors	Authors
Sljivo and Gallina (2016)	Aviation	Wheel breaking system	✓	Authors	Authors

Table 14 (continued)

Study	Domain	Data source	SAC	Creators	Validators
Strielkina et al. (2018)	Medical	Healthcare IoT system		NA	Authors
Tippenhauer et al. (2014)	Electrical	An electrical power grid use case	√	Authors	Authors
Vivas et al. (2011)	Software Engineering	The research project PICOS (Privacy and Identity Management for Community Services)	√	Authors	Authors
Experiment					
Chindamaikul et al. (2014)	Information Retrieval	Open source software development project	√	Expert group	Authors
Gallo and Dahab (2015)	Education	Course in information security	√	NA	Authors*
Masumoto et al. (2013)	Software Engineering	Commercial web application	√	Authors	Authors
Observation					
Finnegan and McCaffery (2014b) (2)	Medical	Technical report	√	Authors	3rd party

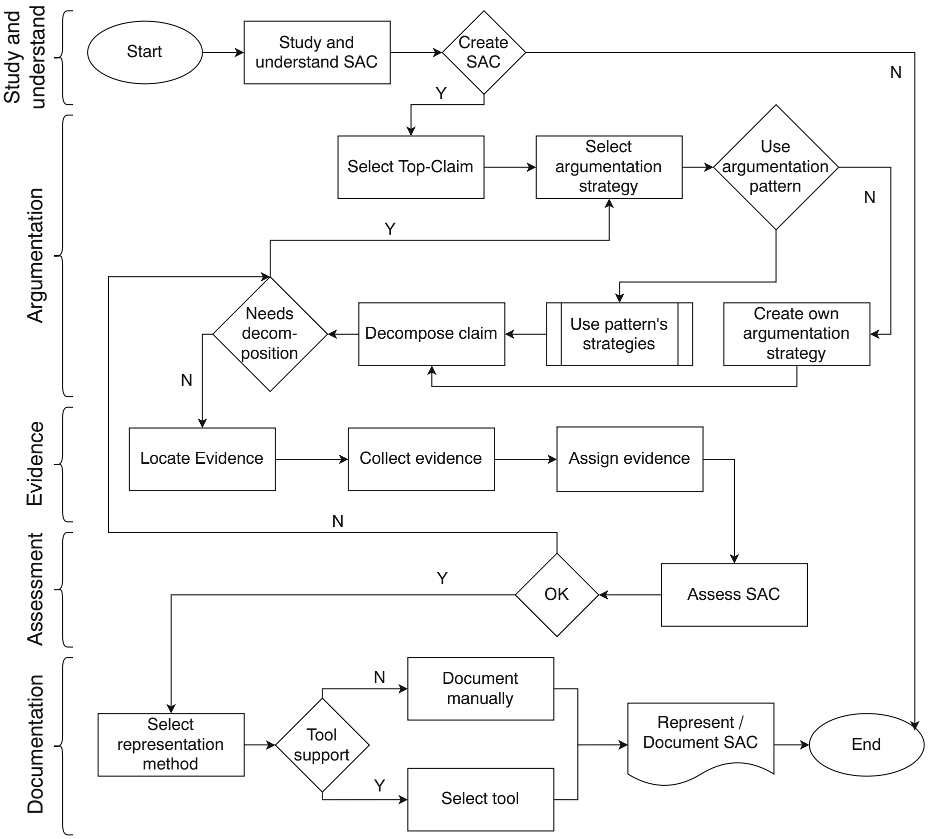

Fig. 5 Flowchart of SAC creation

to use a pattern, and create their own unique structure if the situation requires that. A pattern is created based on the knowledge gathered while creating SACs. It is outside the scope of this workflow. However, this is discussed in the recommended papers in Table 15.

Evidence This block includes locating, collecting, and assigning evidence to the claims of the SAC. In some cases, the evidence is not present when the SAC is being built; hence, they need to be created. In our workflow, this would be a part of the `collect evidence` activity. Moreover, these activities might be done in an iterative manner, including the assessment.

Assessment This block focuses on assessing SACs. This is done to check the quality of the created SAC, and, e.g., to determine whether a claim needs extra evidence to reach a certain confidence level. Assessment starts after the claims have been identified and the evidence is assigned to the corresponding claims. The result of this step might require the creators of the SAC to go back to the point where they assess a claim and make a decision whether or not to further decompose it or assign evidence to it. Since there is a lack of studies that focus on quality assurance of SAC, we have recommended studies which include some metrics to help assessing SACs in Table 15.

Table 15 Recommended reading material for each block of the SAC creation workflow

Block	Recommended reading
Study and understand SAC	Agudo et al. (2009), Alexander et al. (2011), Ben Othmane et al. (2014), Gallo and Dahab (2015), Knight (2015), Netkachova and Bloomfield (2016), Weinstock et al. (2007)
Argumentation	Agudo et al. (2009), Ben Othmane and Ali (2016), Coffey et al. (2014), Hawkins et al. (2015), Mohammadi et al. (2018), Tippenhauer et al. (2014), Vivas et al. (2011), Xu et al. (2017)
Evidence	Cheah et al. (2018), Chindamaikul et al. (2014), Lipson and Weinstock (2008), Shortt and Weber (2015)
Assessment	Chindamaikul et al. (2014), Rodes et al. (2014)
Documentation	Ben Othmane and Ali (2016), Ionita et al. (2017), Poreddy and Corns (2011), Tippenhauer et al. (2014)

Documentation This block includes making a decision of whether or not to use a tool for modelling the argument and documenting the SAC. If a tool is used, then the notation to be used is limited to the one/s supported by the tool. If the documentation will be created manually, the creators will have the freedom to use an existing notation, extend one, or even create their own.

6 Discussion

In this section we discuss the main findings and insights we gathered while reading the papers included in this study. In summary the main observations are the following:

- There are potential benefits of SAC adoption, but further investigation is need.
- There is a rich variety of approaches, with room for improvement.
- Knowledge transfer from the safety domain should take into consideration differences between safety and security.
- There is a lack of quality assurance of the outcomes, which should be avoided in future studies.
- There is imbalanced coverage in literature, which requires more academic research.
- There is room for improvement when it comes to support, which requires companies or the open-source community to step up.
- There is a lack of a mature guidelines for SAC adoption, which might require a standardization activity.

6.1 Potential for a Wide Range of Benefits

The literature is full of motivations for using SAC, as well as suggestions for where to use them, as our results of RQ1 show in Sections 4.2.1 and 4.2.2. We consider this to be a positive factor. However, our impression is that these motivations are on a high level and lack detailed studies to show how realistic and applicable they are. For example, many papers motivate the adoption of SAC as a way to establish compliance with regulation and standards without pinpointing the regulations' and standards' specific constraints for using SAC. Without having an in-depth knowledge of the specific regulation or standard, it is hard to determine whether SACs are explicitly required, or rather just recommended as a way to create a structured argument for security. Incidentally, in our own previous work we have tried to demystify this issue in the context of automotive systems (Mohamad et al. 2020).

Furthermore, some studies suggest that SACs can be used for evaluating the level of security of a system by assigning measurements to the elements of the cases, e.g., the evidence. However, these studies lack detailed guidelines of how to create these quantitative attributes. An example is the usage scenario that suggests to use attribute values for evaluating an architecture (Yamamoto 2015). The approach suggests to assign values to evidence, ranging from −2 to 2, based on how satisfying the evidence are to the claims, i.e., to what degree the provided evidence justify the claims. However, there is no specific criteria for determining the attribute value, making the exercise subjective and nontransparent.

Another example is when SACs are suggested as tools to aid in information security education (Gallo and Dahab 2015). This very interesting concept is not supported by a discussion on the required level of detail in the SACs presented to students.

We believe that there is a substantial gap between the potential of SAC reported in literature and their application in industry. An obvious question to ask is: why are SACs not more widely adopted in industry even though there are so many motivations and usage scenarios for them in literature? It has already been shown that adopting SACs is nontrivial (Mohamad et al. 2020). It requires a substantial amount of effort and time, which grows as the systems become more complex. It also comes with many challenges, such as finding the right expertise to create them. Furthermore, the challenges do not stop at the creation of SACs, but are extended to updating, maintaining, and making them accessible at the right level of abstraction to the right users. We believe that these matters need to be addressed in studies that suggest the usage of SACs in different domains.

6.2 Wide Variety of Approaches

The literature includes a rich variety of studies which explore approaches for creating SACs, especially when it comes to the argumentation part, as shown in the results of RQ2 in Section 4.3. This gives organizations the possibility to choose those approaches that fit their way of working and the security artefacts they produce. For example, a company that works according to an agile methodology could choose to adopt an SAC approach for iterative development (Ben Othmane and Ali 2016). However, this choice has to consider constraints of the applicability of the approach, including benefits and challenges of its adoption. These aspects are not discussed in the literature and the burden is left to the adopter.

Another example is the question of conformance with different standards. While this has been discussed in literature, there is a lack of studies which systematically assess different approaches based on their ability to help achieving conformance with a certain standard. To generalize this, we observed that there is a lack of studies which compare different approaches in different contexts. In consequence, from an industrial perspective, organizations need to select suitable approaches in an exploratory way, which can be confusing.

The studies presenting new approaches also lack the discussion of the granularity level that is possible, or required to achieve using each approach. We believe that future studies should take into consideration the possible usages for SACs created using different approaches, and discuss the required granularity level based on that. For example, would a

SAC created through the security assurance-driven software development approach (Vivas et al. 2011) be useful to companies which outsource parts of their development work to providers? In that case, on which level should these cases be created, e.g., on the feature level or on the level of the complete product?

Lastly, we believe that there is room for exploration of hybrid approaches which combine two or more of the approaches reported in literature. This becomes especially important when different approaches target individual parts of SAC, e.g., argumentation and evidence.

6.3 Security Might Differ from Safety

We have seen in many cases that the approaches presented in literature treat security and safety cases as the same, e.g., Chindamaikul et al. (2014), Graydon and Kelly (2013), Hawkins et al. (2015), Sljivo and Gallina (2016), Goodger et al. (2012), Fung et al. (2018), Ankrum and Kromholz (2005), Gacek et al. (2014) and Sklyar and Kharchenko (2017b, 2019). We believe that since assurance cases in general are mature in the safety domain and have been used for a long time, it is natural to consider the gained knowledge and transfer it into other domains, such as security. However, this knowledge transfer has to take into consideration the differences between safety and security, e.g., in terms of field maturity and nature. For example in safety, there is usually a wide access to information in contrast to security, where threat and risk analysis are considered sensitive information (Piètre-Cambacédès and Bouissou 2013).

Alexander et al. (2011) provide a discussion on the differences between safety and security both from theoretical, and practical aspects. Other studies combine security and safety assurance by creating combined arguments or security-informed safety arguments (Taguchi et al. 2014; Netkachova and Bloomfield 2016; Cockram and Lautieri 2007; Netkachova et al. 2015). We have also seen that some studies use different types of assurance cases to argue for security in Section 4.3. The results do not show any noticeable differences to SAC. This means that we were not able to find any special characteristics in the different types of ACs that distinguish them from SAC, when they are applied on security. However, the approaches for creating the argumentation part differ among the types according to their focus, e.g., trustworthiness and depend-ability.

6.4 Lack of Quality Assurance

Quality assurance is the weaker part of the literature reviewed in this study. We talk here about three main things. First is the quality of the outcomes when it comes to their applicability in practice. We have seen in the results of RQ4 in Section 4.5 that *illustrative cases* are often preferred over types of more empirically grounded validation. This indicates scarcity of industrial involvement. The reason might be a lack of interest, which contradicts with the reported motivations and usage scenarios, or simply because it is hard to get relevant data from industrial companies to validate the outcomes, as security-related data is considered to be sensitive (as we mentioned earlier). Furthermore, with the exception of a few cases, the creation and validation of SAC in literature is done by the authors of the studies. We believe that this contributes heavily to the lack of information addressing challenges and drawbacks of applying SACs in a practical context.

The second issue is the generalize-ability of the approaches with regards to their used argumentation strategies. The approaches we reviewed use a wide variety of argumentation strategies, e.g., based on threat analysis, requirements, or risk analysis. However, they lack validations and critical discussions as to whether the approaches work only with the used strategies or can use other strategies as well. We suggest to validate these approaches based on different types of strategies in future research.

The last point is the lack of mechanisms for building-in quality assurance within the SACs. We believe that it is essential for the argumentation provided in SACs to be complete in order for them to be useful. For that there needs to be a mechanism to actively assess the quality of the arguments to gain confidence in them. This is not addressed in literature apart from a few studies, e.g., Chindamaikul et al. (2014) and Rodes et al. (2014). Similarly, the evidence part also needs to be assessed. e.g., by introducing metrics to assess the extension to which a certain evidence justifies the claim it is assigned to. The inter-relation between claims and evidence need to be addressed. For example each claim can have a certain saturation level to be achieved, and each evidence provides a degree of saturation. Hence, it would be possible to assess whether the claim is fully satisfied or not by the assigned evidence.

6.5 Imbalance in Coverage

The coverage of matters related to SAC in literature is imbalanced to a large extent. When it comes to the approaches, our results in Section 4.3.1 indicate a tendency towards covering the argumentation part more than the evidence part. This indicates a weakness in the approaches, as elements of SAC cannot be evaluated in silos. For example, if we take an approach to create security arguments, how would we know which evidence to associate with these. Moreover, we will not be able to assess whether we actually reach an acceptable level of granularity for the claims to be justified by evidence. Same thing applies for the evidence part. If we only look at the evidence we will not be able to know which claims the suggested evidence can help justify. To be able to evaluate the evidence, they have to be put in context with the rest of the SAC. When reviewing the studies that focus on one element of SAC, we were not able to find any links to related studies focusing on the remaining elements, which indicates incompleteness of the approaches especially for putting them into practice.

When it comes to other areas, the assessment and quality assurance of SAC is rarely covered, as we discussed in the previous sub-section. Furthermore, there is a lack of studies covering what comes after the creation of SAC. In particular, for SAC to be useful, they have to be updated and maintained throughout the life-cycles of the products and systems they target, otherwise, they become obsolete (Mohamad et al. 2020). Particularly, there need to be traceable links between the created SACs and the artefacts of these products and systems. Many SAC approaches use GSN, which allows to reference external artefacts using the context and assumption nodes. However, these nodes are rarely exploited in the examples provided in the studies we reviewed.

6.6 Room for Support Improvement

The tools reported in literature cover activities related to AC, such as creation, documentation and visualization, as shown in the results of RQ3 in Section 4.4. Some of these tools have features such as the validation of AC based on consistency rules related to the used notation, or even user-specified rules (Adelard 2003). Other tools assist in the maintenance of AC through change impact analysis (Fung et al. 2018), and assessment of AC (G.U. of Technology 2010). When it comes to automatic creation of SAC, there were only coverage for the argumentation part (Tippenhauer et al. 2014; Hawkins et al. 2015), which reflects the imbalance in coverage we discussed earlier.

What we observed is that most of the tools are originally created for supporting safety cases, and not in particular SAC. As a consequence, they lack specific features which can be very helpful while building SAC. In particular, we note the fact that security assurance cases need to be treated as living documents (more so than their safety counterparts) due to a continually shifting threat landscape. For example, there is no tool that integrates with other security tools, e.g., an intrusion detection system, to actively update evidence. In general, we note that the tools lack integration with other systems, which agrees with what Maksimov et. al. reported in their study (Maksimov et al. 2018).

Moreover, even though some studies have reported the demonstration of created tools using a case study, e.g., Tippenhauer et al. (2014), it is not clear how flexible they are to be tailored for specific needs of a certain organization, and to be integrated with their tool-chain. We believe that in order for practitioners to use these tools, there needs to be a certain amount of confidence, which is absent due to little reported usage or replications in industry. The same thing applies for the reported patterns. For a specific artefact to be qualified as a pattern, it needs to be used in several studies and in several contexts, which is not the case. Additionally, as we discussed earlier, some important aspects, e.g., traceability is not covered in literature, and this is also the case when it comes to the reported supporting tools.

We also believe that there is room for creativity in the development of the tools. For instance, there are no supporting tools which use machine learning techniques to predict whether a requirement or test case qualifies to be a part of a SAC. This opens up opportunities for companies and the open-source community to step up and close the gap between the potential and the current support.

6.7 Need for a Guideline

Finally, we believe that there is a need for an explicit guideline for on-boarding a SAC-based approach in an industrial context. We believe that with the current level of maturity in related literature, companies which want to adopt SAC approaches have to account for a high cost, as they have to learn, experiment and develop a lot internally. This is due to the lack of reported validation and lessons learned from industry, but another sign is the lack of tool support specific for SAC (as mentioned above).

Standardization bodies are aware of the importance of SAC, as they are being mentioned as requirements in some security standards and best practice documents, e.g., the upcoming standard for cyber-security in automotive ISO21434 (International Organization for Standardization and Society of Automotive Engineers 2018). However, these standards do not provide any specific guideline or constraints for how SAC should be created and used. It is important that key players in selected domains (e.g., automotive and healthcare) put together efforts to standardize the scope and requirements related to SAC. We believe that this would elevate the maturity in the field.

7 Validity Threats

In this study, we consider the internal and external categories of validity threats as defined in Campbell and Stanley (2015), and described in Wohlin et al. (2012) and Kitchenham et al. (2007). The work of conducting the review was done by one researcher. This means that applying the inclusion / exclusion criteria in each of the four filtering rounds was done by one person. This imposes a risk of subjectivity, as well as a risk of missing results, which might have affected the internal validity of this study. To mitigate this, a preliminary list of known good papers was manually created and used for a sanity check of the selected and included papers. Additionally, a quality control was performed periodically by the other authors to check the included and excluded studies.

Restricting our search to three digital libraries could have increased the probability of the risk of missing relevant studies. This was mitigated by performing the snowballing search to search for papers that are not necessarily included in the databases of the three considered libraries.

Another threat to validity is publication bias (Kitchenham et al. 2007). This is due to the fact that studies with positive results are more likely to get published than those with negative results. This could compromise the conclusion validity of this SLR, as in our case we did not find any study that is, e.g., against using SAC, or which reported a failed validation of its outcome. In our study, we have partially mitigated this threat by also including a few technical reports (i.e., non peer-reviewed material). These papers have been identified as part of the snowballing, as we did not restrict to peer-reviewed papers.

External validity depends on the internal validity of the SLR (Kitchenham et al. 2007), as well as the external validity of the selected studies. We did scan gray literature to mitigate publication bias, but we excluded studies that are under 3 pages, and old studies as exclusion criteria to mitigate the risk of including studies with high external validity threats.

When it comes to the reliability of the study, we believe that any researcher with access to the used libraries will be able to reproduce the study, and get similar results plus additional results for the studies which get published after the work of this SLR is done.

8 Conclusion and Future Work

In this study, we conducted a systematic review of the literature on security assurance cases. We used three digital libraries as well as snowballing to find relevant studies. We included 51 studies as primary data points, and extracted the necessary data for the analysis.

The main findings of our study show that many usage scenarios for SAC are mentioned, and that several approaches for creating them are discussed. However, there is a clear gap between the usage scenarios and approaches, on one side, and their applicability in real world, on the other side, as the provided validations and tool support are far from being sufficient to match the level of ambition. Based on the results of this systematic literature review, we created a workflow for working with SAC, which is a useful tool for practitioners and also provides a guideline on how to approach the study of the literature, i.e., which paper is relevant in each stage of the workflow.

Based on our results and findings, in the future we will be working to close the gap between research and industry when it comes to applying security assurance cases. We will be looking into exact needs and challenges for these cases in specific domains, e.g., automotive. We believe that introducing SAC in large organizations needs appropriate planning to, e.g., find suitable roles for different tasks related to SAC, and integrating with current activities and way of working. Hence, we see a potential direction of future work in that area.

When it comes to the technical work, we believe that there is room for improvement in the approaches for SAC creation, especially when it comes to the evidence part. For instance, a possible future work direction is to look into ways to automatically locate, collect, and assign evidence to different claims.

Finally, we believe that quality assurance of SAC has not been addressed sufficiently in literature. As a future work, we will look into ways to ensure the completeness of a security case when it comes to the argumentation, as well as the confidence in how well the provided evidence justify these claims.

Acknowledgements This work is partially supported by the CASUS research project funded by VINNOVA, a Swedish funding agency. We sincerely thank three anonymous reviewers whose comments and suggestions helped improve and clarify this manuscript.

Funding Open access funding provided by University of Gothenburg.

References

Adelard (1998) The adelard safety case development manual

Adelard (2003) The adelard safety case editor—asce. Product description available at: http://adelard.co.uk/software/asce/

Agudo I, Vivas JL, López J (2009) Security assurance during the software development cycle. In: Proceedings of the international conference on computer systems and technologies and workshop for PhD students in computing. ACM, p 20

Alexander R, Hawkins R, Kelly T (2011) Security assurance cases: motivation and the state of the art. High Integrity Systems Engineering Department of Computer Science University of York, Deramore Lane York YO10 5GH

Ankrum TS, Kromholz AH (2005) Structured assurance cases: three common standards. In: Ninth IEEE international symposium on high-assurance systems engineering (HASE'05), pp 99–108. https://doi.org/10.1109/HASE.2005.20

Australian Research Council (2018) Excellence in research for Australia. https://www.arc.gov.au/excellence-research-australia

Behrmann G, David A, Larsen KG, Håkansson J, Pettersson P, Yi W, Hendriks M (2006) Uppaal 4.0. In: Behrmann G et al (eds) Uppaal 4.0. Third international conference on the quantitative evaluation of SysTems (QEST 2006). IEEE Computer Society, Los Alamitos

Ben Othmane L, Ali A (2016) 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). IEEE, pp 564–571

Ben Othmane L, Angin P, Bhargava B (2014) Using assurance cases to develop iteratively security features using scrum. In: 2014 Ninth international conference on availability, reliability and security. IEEE, pp 490–497

Birch J, Rivett R, Habli I, Bradshaw B, Botham J, Higham D, Jesty P, Monkhouse H, Palin R (2013) Safety cases and their role in iso 26262 functional safety assessment. In: International conference on computer safety, reliability, and security. Springer, pp 154–165

Bloomfield R, Bishop P (2010) Safety and assurance cases: past, present and possible future–an adelard perspective. In: Making systems safer. Springer, pp 51–67

Bloomfield R, Bishop P, Butler E, Netkachova K (2017) Using an assurance case framework to develop security strategy and policies. In: International conference on computer safety, reliability, and security. Springer, pp 27–38

Calinescu R, Weyns D, Gerasimou S, Iftikhar MU, Habli I, Kelly T (2017) Engineering trustworthy self-adaptive software with dynamic assurance cases. IEEE Trans Softw Eng 44(11):1039–1069

Campbell DT, Stanley JC (2015) Experimental and quasi-experimental designs for research. Ravenio Books

Cheah M, Shaikh SA, Bryans J, Wooderson P (2018) Building an automotive security assurance case using systematic security evaluations. Comput Secur 77:360–379

Chindamaikul K, Takai T, Iida H (2014) Retrieving information from a document repository for constructing assurance cases. In: 2014 IEEE international symposium on software reliability engineering workshops. IEEE, pp 198–203

Cockram T, Lautieri S (2007) Combining security and safety principles in practice. In: Proceedings of the 2nd institution of engineering and technology international conference on system safety. IET, pp 159–164

Coffey JW, Snider D, Reichherzer T, Wilde N (2014) Concept mapping for the efficient generation and communication of security assurance cases. Proc IMCIC 14:173–177

Computing Research and Education Association of Australasia: core ranking portal—computing research and education. https://www.core.edu.au/conference-portal (2018)

Cyra L, Gorski J (2007) Supporting compliance with security standards by trust case templates. In: 2nd International conference on dependability of computer systems (DepCoS-RELCOMEX'07). IEEE, pp 91–98

Easterbrook S, Singer J, Storey MA, Damian D (2008) Selecting empirical methods for software engineering research. In: Guide to advanced empirical software engineering. Springer, pp 285–311

Feiler PH, Gluch DP (2012) Model-based engineering with AADL: an introduction to the SAE architecture analysis & design language. Addison-Wesley

Finnegan A, McCaffery F (2014a) A security argument pattern for medical device assurance cases. In: 2014 IEEE international symposium on software reliability engineering workshops. IEEE, pp 220–225

Finnegan A, McCaffery F (2014b) Towards an international security case framework for networked medical devices. In: International conference on computer safety, reliability, and security. Springer, pp 197–209

Finnegan A, McCaffery F, Coleman G (2013) A process assessment model for security assurance of networked medical devices. In: International conference on software process improvement and capability determination. Springer, pp 25–36

Fung NL, Kokaly S, Di Sandro A, Salay R, Chechik M (2018) Mmint-a: a tool for automated change impact assessment on assurance cases. In: International conference on computer safety, reliability, and security. Springer, pp 60–70

Gacek A, Backes J, Cofer D, Slind K, Whalen M (2014) Resolute: an assurance case language for architecture models. ACM SIGAda Ada Lett 34(3):19–28

Gade D, Deshpande S (2015) A literature review on assurance driven software design. Int J Adv Res Comput Commun Eng 4(9):82–87

Gallo R, Dahab R (2015) Assurance cases as a didactic tool for information security. In: IFIP World conference on information security education. Springer, pp 15–26

GessNet (2011) Turboac^{TM} assurance cases. https://www.gessnet.com//

Goodger A, Caldwell N, Knowles J (2012) What does the assurance case approach deliver for critical information infrastructure protection in cybersecurity? In: 7th IET International conference on system safety, incorporating the Cyber security conference. IET

Górski J, Jarzębowicz A, Miler J, Witkowicz M, Czyżnikiewicz J, Jar P (2012) Supporting assurance by evidence-based argument services. In: International conference on computer safety, reliability, and security. Springer, pp 417–426

Graydon PJ, Kelly TP (2013) Using argumentation to evaluate software assurance standards. Inf Softw Technol 55(9):1551–1562

Group GCSW (2011) Gsn community standard. Available at www.goalstructuringnotation.info/

G.U. of Technology (2010) Nor-sta. https://www.nor-sta.eu/en/

Haley CB, Moffett JD, Laney R, Nuseibeh B (2005) Arguing security: validating security requirements using structured argumentation. In: Proceedings of the 3rd symposium on requirements engineering for information security (SREIS'05)

Hawkins R, Habli I, Kolovos D, Paige R, Kelly T (2015) Weaving an assurance case from design: a model-based approach. In: 2015 IEEE 16th international symposium on high assurance systems engineering. IEEE, pp 110–117

He Y, Johnson C (2012) Generic security cases for information system security in healthcare systems. In: 7th IET international conference on system safety, incorporating the Cyber security conference. IET

International Organization for Standardization (2011) ISO 26262 Road vehicles—Functional safety, 1st edn

International Organization for Standardization and Society of Automotive Engineers (2018) ISO/SAE 21434 Road vehicles—Cybersecurity Engineering, CD Draft

Ionita D, Kegel R, Baltuta A, Wieringa R (2016) Arguesecure: out-of-the-box security risk assessment. In: 2016 IEEE 24th international requirements engineering conference workshops (REW), pp 74–79. https://doi.org/10.1109/REW.2016.027

Ionita D, Ford M, Vasenev A, Wieringa R (2017) Graphical modeling of security arguments: current state and future directions. In: International workshop on graphical models for security. Springer, pp 1–16

Kitchenham B et al (2007) Guidelines for performing systematic literature reviews in software engineering. Tech. Rep. EBSE-2007-12007 Keele University

Knight J (2015) The importance of security cases: proof is good, but not enough. IEEE Secur Privacy 13(4):73–75

Lipson H, Weinstock C (2008) Evidence of assurance: laying the foundation for a credible security case. Tech. rep., Carnegie Mellon University

Maksimov M, Fung NL, Kokaly S, Chechik M (2018) Two decades of assurance case tools: a survey. In: International conference on computer safety, reliability, and security. Springer, pp 49–59

Maksimov M, Kokaly S, Chechik M (2019) A survey of tool-supported assurance case assessment techniques. ACM Comput Surv 52(5). https://doi.org/10.1145/3342481

Masumoto M, Tokuno T, Yanamoto S (2013) A method for assuring service grade with assurance case: An experiment on a portal service. In: 2013 IEEE international symposium on software reliability engineering workshops (ISSREW). IEEE, pp 311–314

Matsuno Y, Takamura H, Ishikawa Y (2010) A dependability case editor with pattern library. In: 2010 IEEE 12th international symposium on high assurance systems engineering. IEEE, pp 170–171

Mohamad M, Åström A, Askerdal O, Borg J, Scandariato R (2020) Security assurance cases for road vehicles: an industry perspective. In: Proceedings of the 15th international conference on availability, reliability and security, ARES '20. Association for Computing Machinery, New York. https://doi.org/10.1145/3407023.3407033

Mohammadi NG, Ulfat-Bunyadi N, Heisel M (2018) Trustworthiness cases–toward preparation for the trustworthiness certification. In: International conference on trust and privacy in digital business. Springer, pp 244–259

Nair S, de la Vara JL, Sabetzadeh M, Briand L (2013) Classification, structuring, and assessment of evidence for safety–a systematic literature review. In: 2013 IEEE sixth international conference on software testing, verification and validation. IEEE, pp 94–103

Netkachova K, Bloomfield RE (2016) Security-informed safety. Computer 49(6):98–102

Netkachova K, Bloomfield R, Popov P, Netkachov O (2014) Using structured assurance case approach to analyse security and reliability of critical infrastructures. In: International conference on computer safety, reliability, and security. Springer, pp 345–354

Netkachova K, Müller K, Paulitsch M, Bloomfield R (2015) Investigation into a layered approach to architecting security-informed safety cases. In: 2015 IEEE/AIAA 34th digital avionics systems conference (DASC). IEEE, pp 6B4–1

Object Management Group (OMG) (2020) Structured assurance case metamodel (SACM), version 2.1. OMG Document Number formal/20-04-01 (https://www.omg.org/spec/SACM/2.1/PDF)

Palin R, Ward D, Habli I, Rivett R (2011) Iso 26262 safety cases: compliance and assurance. In: 6th IET international conference on system safety. IET

Patu V, Yamamoto S (2013a) How to develop security case by combining real life security experiences (evidence) with d-case. Procedia Comput Sci 22:954–959

Patu V, Yamamoto S (2013b) Identifying and implementing security patterns for a dependable security case–from security patterns to d-case. In: 2013 IEEE 16th international conference on computational science and engineering. IEEE, pp 138–142

Piètre-Cambacédès L, Bouissou M (2013) Cross-fertilization between safety and security engineering. Reliab Eng Syst Saf 110:110–126. https://doi.org/10.1016/j.ress.2012.09.011. http://www.sciencedirect.com/science/article/pii/S0951832012001913

Poreddy BR, Corns S (2011) Arguing security of generic avionic mission control computer system (mcc) using assurance cases. Procedia Comput Sci 6:499–504

Ray A, Cleaveland R (2015) Security assurance cases for medical cyber–physical systems. IEEE Des Test 32(5):56–65

Rodes BD, Knight JC, Wasson KS (2014) A security metric based on security arguments. In: Proceedings of the 5th international workshop on emerging trends in software metrics. ACM, pp 66–72

Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. Empir Softw Eng 14(2):131

Sein M, Henfridsson O, Purao S, Rossi M, Lindgren R (2011) Action design research. MIS Q 35:37–56. https://doi.org/10.2307/23043488

Shortt C, Weber J (2015) Hermes: a targeted fuzz testing framework. In: International conference on intelligent software methodologies, tools, and techniques. Springer, pp 453–468

Singapore ADSC (2015) Cybersage https://www.illinois.adsc.com.sg/cybersage/index.html/

Sklyar V, Kharchenko V (2016) Assurance case driven design for computer systems: graphical notations versus mathematical methods. In: 2016 Third international conference on mathematics and computers in sciences and in industry (MCSI). IEEE, pp 308–312

Sklyar V, Kharchenko V (2017a) Challenges in assurance case application for industrial iot. In: 2017 9th IEEE international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS), vol 2. IEEE, pp 736–739

Sklyar VV, Kharchenko VS (2017b) Assurance case driven design based on the harmonized framework of safety and security requirements. In: ICTERI, pp 670–685

Sklyar V, Kharchenko V (2019) Green assurance case: applications for internet of things. In: Green IT engineering: social, business and industrial applications. Springer, pp 351–371

Sljivo I, Gallina B (2016) Building multiple-viewpoint assurance cases using assumption/guarantee contracts. In: Proccedings of the 10th European conference on software architecture workshops. ACM, p 39

Spriggs J (2012) GSN-the goal structuring notation: a structured approach to presenting arguments. Springer Science & Business Media

Strielkina A, Illiashenko O, Zhydenko M, Uzun D (2018) Cybersecurity of healthcare iot-based systems: regulation and case-oriented assessment. In: 2018 IEEE 9th international conference on dependable systems, services and technologies (DESSERT). IEEE, pp 67–73

Taguchi K, Souma D, Nishihara H (2014) Safe & sec case patterns. In: International conference on computer safety, reliability, and security. Springer, pp 27–37

Tippenhauer NO, Temple WG, Vu AH, Chen B, Nicol DM, Kalbarczyk Z, Sanders WH (2014) Automatic generation of security argument graphs. In: 2014 IEEE 20th pacific rim international symposium on dependable computing. IEEE, pp 33–42

Toulmin SE (2003) The uses of argument. Cambridge University Press, Cambridge

Vivas JL, Agudo I, López J (2011) A methodology for security assurance-driven system development. Requir Eng 16(1):55–73

Weinstock CB, Goodenough JB, Lipson HF (2007) Arguing security-creating security assurance cases. Tech. rep., Software Engineering Institute—Carnegie Mellon University. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=293629. Part of the collection "Resources for Assurance Cases"

Willadsen K (2011) Meld. https://meldmerge.org/

Wohlin C (2014) Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering. Citeseer, p 38

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer Science & Business Media

Xu B, Lu M, Zhang D (2017) A layered argument strategy for software security case development. In: 2017 IEEE international symposium on software reliability engineering workshops (ISSREW). IEEE, pp 331–338

Yamamoto S (2015) Assuring security through attribute gsn. In: 2015 5th International conference on IT convergence and security (ICITCS). IEEE, pp 1–5

Yin RK et al (2003) Design and methods. Case Study Research 3

Yu Y, Tun TT, Tedeschi A, Franqueira VNL, Nuseibeh B (2011) Openargue: supporting argumentation to evolve secure software systems. In: 2011 IEEE 19th international requirements engineering conference, pp 351–352. https://doi.org/10.1109/RE.2011.6051671

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mazen Mohamad received his master's degree in software engineering in 2016 from Chalmers University of technology in Sweden, and is currently working towards a PhD at the Software Engineering division of the Computer Science and Engineering department of Chalmers and University of Gothenburg. His research interests include security assurance of cyber-physical systems.

Jan-Philipp Steghöfer is an associate professor at the Software Engineering Division of Chalmers University of Technology and the University of Gothenburg. He studies software traceability in all of its facets and is one of the drivers behind Eclipse Capra, an open source traceability management tool. Jan-Philipp has also worked on safety assessment and on agile software development in the automotive, medical, and avionics domain.

Dr. Riccardo Scandariato received his PhD in Computer Science in 2004 from Politecnico di Torino, Italy. In his academic career he had the opportunity to work in several countries, including the United States (University of Virginia, 2003), Italy (Politecnico di Torino, 2004–2005), Belgium (KU Leuven, 2006–2014) and Sweden (University of Gothenburg, 2014–2020). Since late 2020, he is the head of the Institute of Software Security at the Hamburg University of Technology (TUHH), in Germany. His work focuses on the design ofsecure applications.

Affiliations

Mazen Mohamad[1] · **Jan-Philipp Steghöfer**[1] · **Riccardo Scandariato**[2]

Jan-Philipp Steghöfer
jan-philipp.steghofer@cse.gu.se

Riccardo Scandariato
riccardo.scandariato@tuhh.de

[1] Department of Computer Science and Engineering, University of Gothenburg and Chalmers University of Technology, Gothenburg SE-41296, Sweden

[2] Institute of Software Security, Hamburg University of Technology (TUHH), Blohmstraße 15, 21079 Hamburg, Germany