Increasing the Confidence in Security Assurance Cases using Game Theory

Increasing the Confidence in Security Assurance Cases using Game Theory

Antonia Welzel
welzel@chalmers.se
Chalmers | University of Gothenburg
Gothenburg, Sweden

Rebekka Wohlrab
wohlrab@chalmers.se
Chalmers | University of Gothenburg
Gothenburg, Sweden

Mazen Mohamad
mazenm@chalmers.se
Chalmers | University of Gothenburg
Gothenburg, Sweden

ABSTRACT

Security assurance cases (SACs) consist of arguments that are supported by evidence to justify that a system is acceptably secure. However, they are a relatively static representation of the system's security and therefore currently not effective at runtime which make them difficult to maintain and unable to support users during threats. The aim of this paper is to investigate how SACs can be adapted to become more effective at runtime and increase confidence in the system's security. We extend an example SAC with game theory, which models the interaction between the system and attacker and identifies their optimal strategies based on their payoffs and likelihoods. The extension was added as a security control in the assurance case, where a security claim indicates what strategy should be taken at runtime. This claim changes dynamically with the recommended strategy output by the game-theoretic model at runtime. Based on the results of the evaluation, the extension was considered to be potentially effective, however this would further depend on how it is implemented in practice.

CCS CONCEPTS

• **Security and privacy → Software security engineering**.

KEYWORDS

Security, Assurance Cases, Runtime, Game Theory, Bayesian Games

ACM Reference Format:
Antonia Welzel, Rebekka Wohlrab, and Mazen Mohamad. 2024. Increasing the Confidence in Security Assurance Cases using Game Theory. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3664476.3664501

1 INTRODUCTION

The security of a system has become increasingly important with higher levels of connectivity and subsequently more extreme consequences of cyberattacks [21]. Therefore, providing assurances for the security of a system is becoming a more essential part of system development and maintenance. One way to assess and ensure system security is using security assurance cases (SACs). They consist of "a structured set of arguments and a corresponding body of evidence to demonstrate that a system satisfies specific claims with respect to its security properties" [26]. SACs can be used to identify potential weaknesses as well as evaluate and maintain the system's compliance with standards and regulations [19]. Hence, they increase the confidence in the system's security, which in this context is defined as the belief that the system's security is reliable. SACs consist of a set of arguments and evidence which build a case describing why the system can be considered acceptably secure. An example SAC of a fictional banking application is shown in Figure 1. For each argument, there is the overall claim that a system is acceptably secure, which branches into different sub-claims and ends in evidence to show that the claims are true [15].

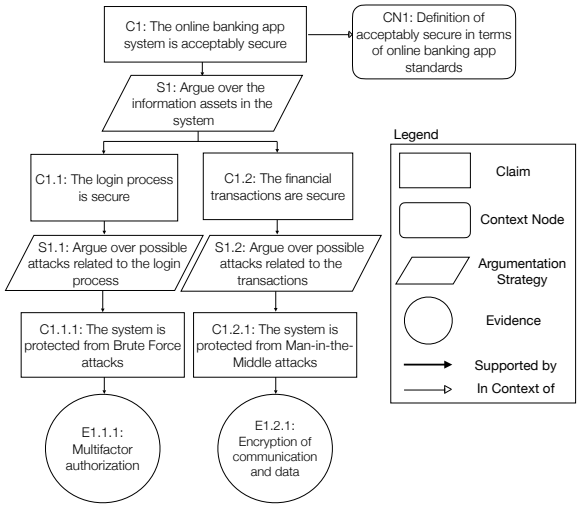

Figure 1: Example of a SAC

SACs are a static representation of the system's security at design time [9]. However, security is dynamic and there will be changes to the system at runtime, which have to be reflected in SACs to maintain their claims and evidence, and consequently, the confidence they provide for the system in question [13]. Additionally, the need for continuous compliance with standards, e.g. ISO/SAE-21434 [8], also requires security analyses of the system at runtime. Therefore, SACs need to capture how potential security incidents affect the system, so that the SACs can continuously reflect its current state and support security teams' decision-making at both design time and runtime.

In this paper, we aim to increase the effectiveness of SACs at runtime and thereby the confidence they can inspire by using game theory. Game theory studies the interactions between independent

ARES 2024, July 30–August 02, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3664501

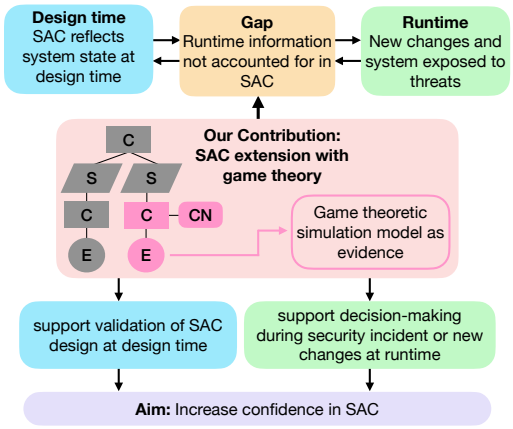

Figure 2: Overview of SAC Extension

self-interested agents, where the actions of each actor are modeled to understand how they will affect each other and what the outcomes might be [11]. It has been applied in computer security to, for instance, model different attack or defense strategies such as in attack-defense models [1, 14, 16, 30]. We attempt to extend SACs with game theory. As shown in Figure 2, the extension incorporates game theory as evidence, which models the level of security in the system while considering the uncertainties of unknown events that might occur at runtime. Consequently, SACs could become more dynamic which would facilitate their design and maintenance as well as potentially enable their use for dynamic decision-making during unexpected behavior at runtime.

In order to approach a solution towards SACs becoming more of a 'living' artifact to support their maintenance and the decision-making for unexpected changes at runtime, we identify the main challenges with maintaining SACs and how they are currently not effective at runtime. Moreover, we explore how SACs can be extended with game theory to evaluate multiple strategies and consider the uncertainties in the environment at runtime. We propose a game-theoretic extension for SACs to make them more dynamic and effective at runtime which is demonstrated on a realistic SAC example. Hence, we answer the following research questions,

> **RQ 1:** What are the current challenges with maintaining SACs at runtime?
> **RQ 2:** To what extent can the confidence in SACs be increased with game theory-based decision-making?
> **RQ 3:** How can SACs be extended to include runtime behavior using a model based on game theory?

The remainder of the paper is structured as follows: Section 2 discusses related work, Section 3 describes the methodology, and Section 4 presents the research findings which are discussed in Section 5. Lastly, conclusions and future work are given.

2 BACKGROUND AND RELATED WORK

This section describes the theoretical background and related work on SACs and game-theoretic decision-making in the security domain.

2.1 Security Assurance Case Development and Use in Practice

SACs are sets of claims and arguments for which evidence is provided to give information and proof about a system's security. The cases are therefore used to document and assess the security of a system as well as how well it fits with the standards and regulations within its industry [17]. Moreover, the use of SACs is mandated by security standards such as the ISO/SAE-21434 [8] from the automotive domain which provides requirements for how SACs should be created and managed to ensure automotive system security.

For each claim in the SAC, there is a set of arguments to show that the claim holds for the system with the support of related evidence [20]. This then increases the confidence in the system that it is secure or "acceptably secure" [9] which refers to the claims and evidence satisfying their security requirements and providing sufficient confidence in the system security based on the current information and context. SACs' argument structures can follow different approaches, such as being based on assets, security requirements, threats, or attack paths, which specify what system properties are in focus for a set of arguments [17].

In practice, the implementation and maintenance of SACs can be relatively complex and vary greatly between industries due to, for example, different regulations or guidelines. Mohamad et al. [18] conducted a study on the use of SACs in the automotive domain and the constraints as well as needs that SACs have to account for. Among the constraints are, for instance, requirements originating from standards that explicitly or implicitly require SACs to include specific evidence and argumentation to ensure acceptable levels of security. Moreover, the internal needs of the organizations in the study related to areas such as alignment of SACs between the companies' internal processes as well as between stakeholders in the supply chain. Additionally, Mohamad et al. [18] identified several use cases to further illustrate the needs of stakeholders in the automotive domain.

2.2 Security Assurance Cases at Runtime

The evidence in SACs is updated or extended with information collected at runtime to be able to maintain confidence in the claims made [5]. Runtime verification can help verify that the assumptions for the SAC are true and provides evidence and assurance for the system's security at runtime [23]. However, the verification often takes place after the system is deployed or in a test environment with conditions similar to deployment and does not enable any knowledge of the system at runtime beforehand. Runtime certification on the other hand is a form of proactive anticipation of a system's runtime behavior, where assumptions in SACs are monitored to predict possible issues. For instance, Rushby [24] uses Bayesian Belief Networks to represent SACs and give an estimate of the confidence of an assumption made at design time and how secure it can be assumed to be at runtime.

In previous research, there have been attempts to make assurance cases more dynamic, such as in the safety domain. For instance, Asaadi et al. [2] propose dynamic assurance cases with an assurance measure based on a dynamic Bayesian network, which provide "operational situational awareness to humans" as well as runtime certification at design time. However, our review of the literature

has shown a scarcity of studies in regards to how specifically SACs can be maintained at runtime and support more continuous maintenance as well as compliance.

A type of system where SACs have to be more dynamic to consider runtime behavior is self-adaptive systems. In these systems, the implementation of SACs that can provide confidence in the security of the system's different states has been difficult, since many tasks and behaviors are not set and usually unknown until runtime as they respond to changes in their environment [5]. Calinescu et al. [5] propose a methodology to make safety assurance cases more dynamic based on dynamic assurance case generation. Jahan et al. [9] propose a runtime adaptation for SACs of self-adaptive systems. Changes to the SAC are passed through a set of values or a 'change set' from the system's MAPE-K loop, such as the state of a variable before and after the change as well as the evidence needed to support the change. Achievement weights are then used to assess the satisficing level of the goal and in turn its level of confidence. The weights are based on the impact of a change and how closely related the changed feature is to the main goals of the system. This adaptation operates at runtime, where it is dependent on what actions the system will actually take. Therefore, it is not a system that anticipates the behavior at design time but rather is dynamic and adapts at runtime. The extension proposed in this paper intends to support decision-making at runtime in response to an attack as well as aid in indicating at design time how the system might operate at runtime, thereby increasing the confidence in the system's security under unknown circumstances.

2.3 Game Theory

Game theory can be used to model the interaction between two independent agents. The main components in games are i) the players, i.e. the actors involved in the game, ii) the actions of the players which can be assumed to be either known or unknown to them, iii) the payoff which is the return for each player in the game based on their actions, and iv) the strategies which are the players' plan of moves in the game for how they will try to win [12]. Depending on the players' strategies and their motivations, there are different types of games. For instance, games can be differentiated as being complete or incomplete information games depending on the players' knowledge of each other's payoffs. For example, in incomplete games there is uncertainty about expected payoffs and therefore players do not have complete information [11]. Additionally, games can be either static or dynamic, where dynamic games are based on multiple moves made by the player and static games only involve one move made by each player which ends the game.

Furthermore, players can adopt different strategies. A pure strategy means choosing one of the actions available to the player [10]. A mixed strategy extends the pure strategy with probability distributions. It reflects one player's uncertainty over the other player's move in reaction to their own and therefore includes more randomness in the action set as the players would randomly choose between the different actions [10]. The best strategy is in the end the one that returns the highest payoff and is identified by the equilibrium of the players' payoffs and strategies [12]. One example of an equilibrium is the Nash equilibrium.

Given that cybersecurity is dynamic and it is influenced by a changing environment, its ability to consider real-time changes is becoming increasingly important. There are multiple studies using game theory to enable more dynamic decision-making in cybersecurity and evaluate risks associated with different threats and mitigations. Game-theoretic decision-making has been applied in network security in the form of attack-defense models. Lye and Wing [14] used a multiplayer game model to simulate interactions between an attacker and a system and to find equilibrium strategies. These security games can be used to model attacks and the defense of a system, thereby assessing its security at runtime. With these attack-defense games the optimal strategies can be identified, either for the attacker or the defense, which makes them useful in aiding decision-making in, for example, system design. Therefore, these games can be effective for both defense analysis and performing security assessments of a system [22].

Security game simulations can be implemented with different types of game-theoretic models depending on the system and the type of interaction between the parties. For instance, a zero-sum stochastic game can uncover the attacker's best strategy using the Nash equilibrium, while also taking into account the uncertainties in the model which makes it more realistic [25]. The Markovian decision process is one stochastic algorithm that has been used to simulate and automate decision-making regarding cybersecurity and defense strategies. McInerney et al. [16] use a Markovian decision process in a single-player game to model a system's defense against attacks. Zheng and Namin [30] present a defense strategy against Distributed Denial-of-Service attacks based on a Markov decision process, where different parameters related to network traffic such as flow entry size are used to represent the states in the model. The model was able to optimize network traffic and detect possible attacks. Moreover, Applebaum et al. [1] used Monte Carlo simulations to simulate possible attacks and defenses as well as identify weaknesses through simulations with the network system, where uncertainty from the environment can then also be taken into account. In this paper, we incorporate the concept of attack-defense models into SACs to enable dynamic decision-making and subsequently increase the cases' effectiveness at runtime.

3 METHODOLOGY

The research aim was to investigate how SACs can be extended to enable dynamic decision support, thereby becoming more runtime effective and increasing the confidence in them. We follow the design science research methodology [6] to develop an extension to SACs (design artifact) to include considerations of their runtime behavior and enable continuous compliance. The design process was split into two iterations. The first iteration focused on the challenges of security assurance at runtime which is related to RQ1. The second iteration focused on extending SACs to become more effective at runtime and enable dynamic decision-making based on game theory, which relates to RQ2 and RQ3. The study's methodology is illustrated in Figure 3.

3.1 Analyzing Challenges

In the first iteration, the challenges of maintaining SACs at runtime were identified by reviewing the literature and validated through

interviews with security experts. These findings provided the general requirements for how the dynamic decision support should be formed to increase confidence in the security assurance at runtime.

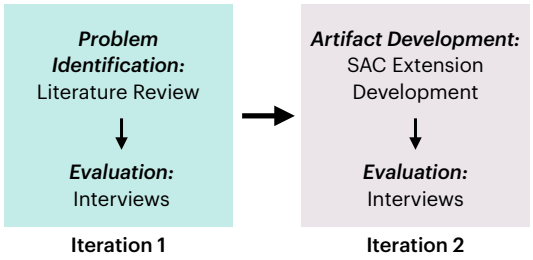

Figure 3: Overview of Methodology

3.1.1 Interviews with Practitioners. Semi-structured interviews with practitioners were performed to gain further insights into the challenges of security assurance at runtime. The participants were chosen based on their knowledge and experience with SACs. It was difficult to find participants who have a sufficiently strong background in cybersecurity and SACs to give meaningful answers. Four interviewees participated, detailed in Table 1, who have extensive knowledge in the area of SACs in practice. They have worked or are currently working in different domains, which also provided insights from multiple contexts.

Table 1: Interviews with Practitioners

ID	Role	Domain	Years of Experience[1]
I1	System Architect	Automotive	15 (5) years
I2	Senior Researcher	Medical, Self-adaptive systems	15 (3) years
I3	Lead Engineer	Automotive	8 (2) years
I4	Process Designer & Architect	Automotive	15 (3) years

[1]Years of experience with SACs in parenthesis

The interviews lasted 45 to 60 minutes and took place digitally or in person. The aim of the interviews was to identify the issues that engineers face when designing security assurance for their systems and how it is impacted by the events at runtime. An interview guide was designed before the interviews [28]. The questions related to security assurance at both design time and runtime, how it affects decision-making as well as how runtime behavior affects the existing security assurance. The questions also covered how automated decision support at runtime for unexpected behavior could be integrated and how it would affect the confidence in SACs. The interviews were recorded with the participant's permission and transcribed. The data were analyzed through a thematic analysis based on the steps by Braun and Clarke [4] described below.
Data Familiarization. The transcriptions were carefully read multiple times to fully understand the content.
Generating Codes. The data were coded with open codes, which enabled an explorative approach to identify the challenges for maintaining SACs at runtime, as they emerge during the coding process and are not set prior to the analysis.
Generating Themes. The codes were grouped based on themes that could be identified in the data.
Reviewing Themes. The themes were discussed and evaluated in a coding workshop among the authors.
Defining Themes. The themes were then further refined and examined based on which interviewees had mentioned an aspect within the theme and their background.
Producing the Report. The results were compiled for the paper.

3.2 Extension Development and Validation

Iteration 2 focused on how SACs can be extended with game theory.

3.2.1 SAC Extension. The research aim was to understand how game theory can be integrated into SACs to increase confidence by providing dynamic decision support and facilitating runtime decision-making during an attack. In order to show how a SAC would need to be adapted, an example SAC was extended with the necessary components to illustrate what the extension would need to include. The example SAC in Figure 1 was used as the basis for the extension. The case is a simple representation of a fictional mobile banking application.

3.2.2 Validation with Practitioners. After the extension had been created, interviews were conducted with each of the security experts from the first iteration to evaluate the extension's potential for increasing the confidence provided in the SAC as well as how effective it would be in a practical context. The interviews lasted around 60 minutes. During each session, the extension was shown and the idea as well as the aim behind it was explained. Additionally, a small-scale simulation of the SAC extension in an attack scenario was shown to illustrate the idea more dynamically and give an example of how it might work in an attack setting. The demonstration was followed up with questions based on an interview guide [27].

The attack scenario that was chosen to be modeled in the simulation was a probe attack performed by the attacker to which a security team can react by either deploying a decoy system or not. The scenario is visualized in the simulation game tree in Figure 6. After showing the artifact, the participants were given a set of seven Likert-scale questions. They aimed to measure three aspects of the extension; its usability, effectiveness, and the confidence it inspires in relation to the SAC, with 1 being the lowest value and 5 being the highest. The closed questions were then followed up with four open-ended questions to enable more discussion around the artifact and identify more specific advantages and issues of the extension for future improvements. The use of questions in both a closed and open-ended format enabled comparability between participants with the closed questions, however also more detailed information with the open questions.

The results of the interviews were afterwards analyzed. For the responses to the closed questions, the central tendency, which included the mean, median, and mode, was calculated and additionally the standard deviation was computed to measure the dispersion of the results. Moreover, a thematic analysis [4] was performed for the open-ended questions according to the steps described in 3.1.1.

3.3 Threats to Validity

In this section, we describe the threats to the research validity.

Internal Validity. Each interview with the security experts was conducted with an interview guide so that the same set of questions were asked. Furthermore, the sampling of participants was based on people who are knowledgeable and experienced in security and SACs. Therefore, this ensures the collection of data that brings useful insights relevant to the research. Moreover, the thematic analysis for the interviews in the first iteration was performed in a coding workshop with all authors which helped reduce the potential bias that might have been introduced in the initial coding.

External Validity. The sample size for the data collection consisted of four security experts, which could be considered rather small and decreases the generalizability of the results. However, they are experts in their field and the goal was not for the participants to be representative of their population and contribute to empirical generalizability, but rather to gather information to make theoretical generalizations such as for the challenges of maintaining SACs at runtime. By ensuring a high quality in the qualitative analysis, generalizations in terms of the theory could still be made. Furthermore, the type of game and strategy to model the runtime interactions between an attacker and the system in the SAC extension was considered carefully to reflect reality as accurately as possible and provide relevant insights. However, the actual game-theoretic part of this model was not in focus for this research and is relatively abstract and simplified.

Construct Validity. The concepts in focus for the interviews, such as security assurance, and how they are used in the research were defined and discussed with the participants before the interview to ensure that the understanding of these concepts matched and to increase construct validity in the data collection.

Reliability. The reliability of the method was maintained through the transparency of the reporting of the research method and findings. For instance, the interview guides are made available.

4 RESULTS

This section describes the challenges of maintaining SACs at runtime and how they can be extended to become effective at runtime.

4.1 Challenges of Maintaining SACs at Runtime

Interviews were conducted to determine the challenges of maintaining SACs at runtime. Seven challenges were identified, shown in Figure 4. There are three challenges occurring at design time and runtime respectively as well as one challenge directly impacting the transition between the two, i.e. the maintenance process.

4.1.1 *Challenges at Design Time.*

System Scope and Boundaries. One challenge is the system scope and boundaries that are defined at design time. During the system design, engineers decide on what requirements the system and its security need to fulfill. All interviewees mentioned that it is difficult to anticipate all scenarios. Consequently, assumptions about the system and its environment are made to define what security features have to be implemented. Making assumptions and defining the scope of security goals and requirements for the system in relation to its context can facilitate the ability to account for possible uncertainties in the defined scope. However, only considering mechanisms within the defined security boundaries makes it difficult to maintain SACs at runtime since there are uncertainties arising from the context that the system is put in that cannot be accounted for. Therefore, while strategic design choices resolve part of the issue by limiting uncertainty, they cannot always cover the context and unexpected behavior from outside the system, resulting in problems at runtime.

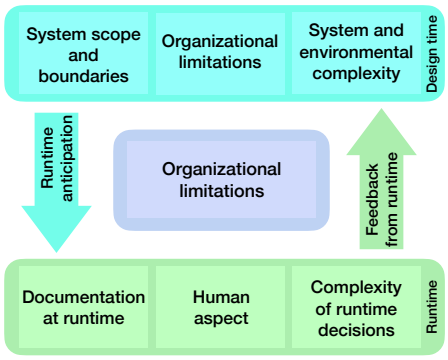

Figure 4: Overview of Challenges for Maintaining Security Assurance at Runtime

Organizational Limitations. Another challenge is presented by the cultural challenges in companies such as a lack of security awareness and culture that can interfere with the maintenance of SACs at runtime. I3 mentioned that too little value is placed on security assurance due to a lack of understanding of it. Oftentimes, testing is viewed as more effective than documentation of the system and the focus on effective security assurance is decreased. Moreover, prioritizing new features over creating more effective security implementations further leads to difficulties with the maintenance of SACs. Additionally, rigid and inflexible organizational structures hinder security assurance design as well as maintenance. Given that security is dynamic, the processes related to security assurance should be flexible to account for changes from, for example, new features, standards, or problems that arise. All interviewees emphasize the importance of organizational fluidity and flexible work structures to make it easier to quickly adapt to changes and that effective processes are a requirement for effective SACs.

System and Environmental Complexity. The system and environmental complexity is another challenge that increases uncertainty and makes it difficult to maintain SACs at runtime. I2 mentioned that since systems are often very complex and operate in complex environments, it is not possible to consider every problem at design time. Moreover, all interviewees stated that at design time it is difficult to plan and account for all possible threats and the effective response due to the dynamic threat landscape and the uncertainty stemming from not knowing who the attacker is. I3 stated that *"the uncertainty comes from who is the attacker going to be. Is it going to be the bored kid in his basement just experimenting with his stuff (...) or is it going to be (...) a major company that wants to take you down a peg and get at your secrets, in which case they are gonna come with a much more well hidden and manicured attack"*. The differences in complexity and resources of attacks affect the

mitigation and the assurance needed at runtime as well as how security assurance is maintained. However, some form of threat analysis typically takes place to get more insights into what might happen at runtime. All interviewees mentioned the importance of verification of the assumptions made at design time and therefore the security assurance at runtime, as well as the necessity of testing in the actual context to get relevant results. However, I3 reported that it is often difficult to test all aspects of the system and thereby also the claims in the security assurance documentation. I2 also brought up the observability issue that exists due to the dynamic nature of systems and their security, where some components of the system and its context can only be observed at runtime and therefore not properly anticipated.

4.1.2 *Challenges at Runtime.*

Documentation at Runtime. One challenge for maintaining security assurance at runtime is their documentation. SACs and security assurance documentation in general are typically very static and are not able to adapt to what happens at runtime. While the complexity of the environment makes it difficult to plan at design time, I2 pointed out that it would also not necessarily be possible to incorporate the dynamic aspects even if they could be anticipated. Consequently, security assurance does not enable quick and flexible decision-making at runtime, which means incident response teams need to respond to problems based on insufficient documentation and this often requires more people to take the right actions leading to a slower response. New evidence is also not dynamically accounted for causing decisions to be less informed. I1 stated, *"the more severe the consequences and impact of a security assurance change are, the more important it becomes to cover the necessary actions at runtime during this kind of incident in the [documentation] to be able to act fast"*.

Human Aspect. Another challenge is the human aspect. I1 mentioned that there is always the possibility for human error both in the analysis of the attack and the decision-making. I2 considered system decision-making, such as in self-adaptive systems, a more reliable option, where human error becomes much more unlikely. However, the actions and scope in these systems are still defined by humans, which also relates to the challenge regarding system scope and boundaries at design time. Therefore, in addition to the potential human error in the analysis at runtime, the decision-making can also be affected by human error in the design process which causes issues at runtime as well.

Complexity of Runtime Decisions. The decisions that have to be made at runtime are often relatively complex since they are dependent on multiple factors as well as often involving many people working in different parts of a system. Decisions tend to move across multiple levels in a company and between different teams, where domain knowledge of security tends to be lost as the decisions are not only influenced by the incident response teams but other engineers with other technical expertise and focus. The complexity in multi-level decision-making as well as the lack of agility in the communication processes slow down the decision-making and the overall ability to quickly respond to attacks. According to I3, SACs are intended to help simplify and facilitate communication, however that is not always the case. Incident response is further impeded when decisions are made with insufficient information

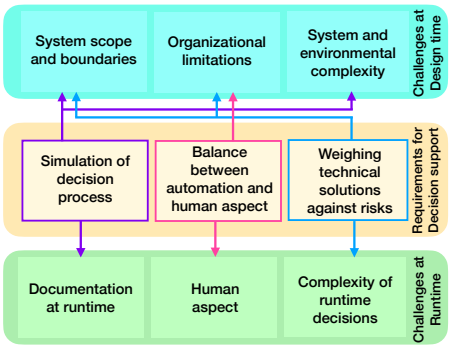

Figure 5: Dynamic Decision Support for Security Assurance

on the relevant contexts of the attack as well as if goals among the decision-makers are not aligned.

4.1.3 *Challenges in the Maintenance Process.*

All interviewees found that there should be a strong interconnection between design time and runtime, which is enabled by continuous maintenance and updates of security assurance as well as effective organizational and workflow structures. However, organizational limitations pose a challenge and hinder maintenance processes, and subsequently the transition of SACs between design time and runtime. This challenge had also been identified at design time as an obstacle to establishing effective structures within companies, but in this context it refers to the issues related to the dynamic feedback loop between the security assurance's states at design time and runtime. According to the interviewees, there is uncertainty around the maintenance processes of SACs. According to I4, most companies are not well organized around these processes leading to slow incorporation of new standards, which require continuous maintenance. Some companies have designated maintenance teams in charge of the changes of SACs and for others, it is the development teams for each product that maintain the security assurance. Furthermore, there is usually a lack of traceability and version control in regards to SACs, which also makes it difficult to maintain them at runtime. Another issue is ineffective communication of updates. I3 stated that the effectiveness of communication is connected to the organization overall and organizational changes often cause communication issues. I1 also mentioned that changes to SACs usually involve multiple forums, which can impact the speed and efficiency of communication. However, this can vary between companies and their workflow structures and processes.

4.2 Requirements for Dynamic Decision Support

The security experts were also asked in the interviews about the requirements for SACs to enable dynamic decision support, shown in Figure 5, and how it could increase their confidence in the assurance case as well as bridge the gap between design time and runtime by addressing some of the previously identified challenges.

4.2.1 *Simulation of Decision Process.*

A dynamic decision support would aid runtime decision-making by simulating runtime behavior and giving an assessment of the impact of a threat. I1 stated that

being able to assess the impact of the threat helps indicate the severity of the issue. They also considered it would be useful if it were able to propose future actions to mitigate a threat. Furthermore, I4 thought that it would be useful if a decision support could visualize the runtime decision processes.

4.2.2 Weighing Technical Solutions against Risks. To increase the confidence in SACs at runtime, the interviews revealed that it is important that a dynamic decision support can help guide the strategy and solutions for runtime decision-making while considering their security risks. I3 reported that the process and discussions regarding what decisions and implementations to make would be facilitated if a dynamic decision support can highlight the risks of different solutions. Moreover, according to I2, a more automated decision support could help compile knowledge about the operational conditions and environment to help anticipate runtime behavior and guide decision-making at design time.

4.2.3 Balance between Automation and Human Aspect. Human errors can arise both during system design as well as runtime analysis and decision-making. Therefore, a more automated decision support would increase the robustness of security-related decision-making and help reduce human error. However, I3 stated that it is still important to make the process understandable for humans and preserve the transparency of a decision support so that humans can interpret it and maintain confidence in the system.

4.3 SAC Extension with Game Theory

The SAC extension proposed in this paper consists of extending a SAC with game theory. The aim of the extension is to increase the confidence in SACs at runtime by, for one, providing a form of validation of design time implementations in terms of how secure the system would be at runtime. Secondly, it should also work as a decision support for humans during an attack at runtime, which can suggest mitigation strategies based on what actions the security team can take to defend the system against the current and potential actions of the attacker. The game-theoretic model is introduced as evidence to a dynamic claim that states the ideal mitigation strategy to take during a security incident. The design of the extension was based on the requirements for dynamic decision support in SACs described in Section 4.2.

Section 4.3.1 presents the example game-theoretic simulation model in our paper and Section 4.3.2 details the SAC extension and how the game theory model was incorporated.

4.3.1 Game-theoretic Model. The example game-theoretic model used to demonstrate the SAC extension in this research is a Bayesian game with mixed strategies. The example attack used to represent the actions by both players, i.e. the system and attacker, is a probe attack that can then be mitigated by either deploying a decoy system or not doing anything, i.e. maintaining the current or 'real' system. The simulation model of the game is shown in Figure 6. After the system has made its move, there are different possible payoffs depending on the type of attacker and their strategies. The payoffs are shown in the boxes at the bottom in Figure 6. The optimal strategy is based on what the payoffs for the system are when taking different actions as well as the probabilities of what actions the other player might take. The ideal strategy is the one with the highest expected payoff. The mixed strategies of the players are illustrated by the different probabilities. In the example in Figure 6, these are set to either 0.3 or 0.7. This means that with 30% or 70% probability, the attacker stops or continues their attack. Depending on the attacker type, the probabilities for the actions vary, which is further described below.

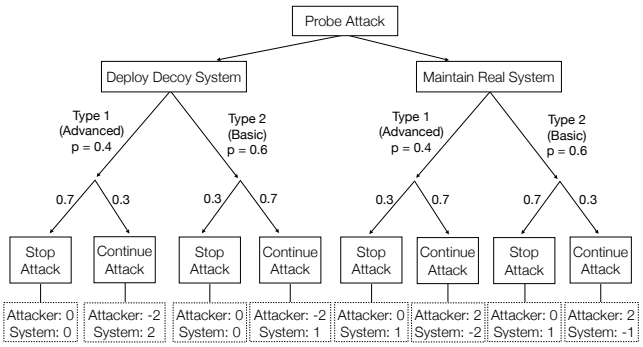

Figure 6: Game-Theoretic Model in the SAC Extension

Due to the uncertainty of what the actual attack might be as well as who the attacker is, there is a high degree of incomplete information that both players have. Neither of them knows what the other party is aware of and what exact strategies or actions they can take. The typical incomplete information game is a Bayesian game as different scenarios can be evaluated based on their likelihood of taking place. The different scenarios are illustrated by a move from another player representing 'Chance', which would be the events taking place outside the players' control to which they can then react with their available strategies. In Figure 6, the game considers two types of attackers with different likelihoods of that scenario being the reality, in this case a 40% and 60% chance for each, which then also influence what the expected payoff would be. The Bayesian game presented here is also a dynamic game, which enables the model to be updated when new information is received about the attacker such as through their chosen moves that would impact the probabilities for different scenarios or actions as well as the potential payoffs of certain outcomes.

Overall, it is difficult to be able to anticipate all possible strategies that an attacker might take and therefore this also impacts the model's ability to identify the optimal strategy. Moreover, the payoffs and probabilities included in the game simulation need to be estimated or based on data models to predict values, which is still an issue that prevents the effectiveness of most attack-defense models and game-theoretic models in general. The system and its environment can be relatively complex and subjected to multiple types of uncertainties, particularly regarding security, that are difficult to account for in SACs. Therefore, the game-theoretic model in the extension would be able to account for some of these uncertainties, which is achieved with probabilities to be able to incorporate the possibilities of different scenarios.

4.3.2 SAC Extension. The proposed extension to the SAC is shown in Figure 7. C3.1. is the overall claim representing the decision support, which is connected to the argumentation strategy arguing

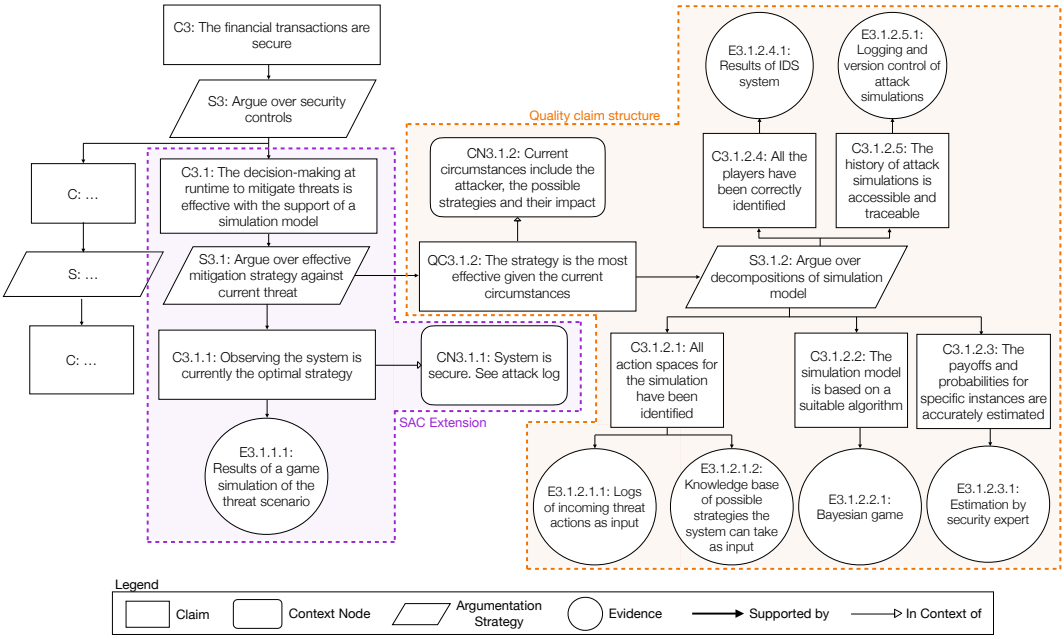

Figure 7: SAC Extension: System is Secure

over the security controls of the system's asset, in this case the financial transactions. It therefore extends the existing security controls, which are either processes or assets to keep the system secure and would also include the different mitigation strategies that the decision support can suggest. C3.1. is followed up with the argumentation strategy for what actions towards threat mitigation should be taken. This argumentation strategy then decomposes into the different claims about what optimal action or defense strategy the security team should take during a threat at runtime.

In Figure 7, the system is assumed to be secure and therefore the current strategy would be to observe the situation, as shown in C3.1.1.. The claim results in the evidence E3.1.1.1., which is the result of the game-theoretic simulation model, described in Section 4.3.1, that calculates what strategy to take. Moreover, C3.1.1. connects to context node CN3.1.1., where the user is referred to a hypothetical attack log that would be able to indicate that there are no incidents that a security team would need to take action against. In order to define and argue for the quality of the extension and the claims being made based on the game-theoretic model, the quality claim QC3.1.2. [19] is connected to argumentation strategy S3.1.2. It justifies why the claims regarding the ideal strategy at runtime are acceptable with the evidence of the simulation model. The quality claim is further decomposed to demonstrate the specific aspects and quality attributes of the simulation model that correlate with the main components of game theory such as the identification of the potential strategies and an adequate estimation of payoffs.

Apart from the quality claim, the claims connected to S3.1. are set up as dynamic claims that will change to the output of what strategy is ideal based on what the game-theoretic simulation model proposes. Dynamic claims would make it easier for the user to follow what action needs to be taken to mitigate a threat. They

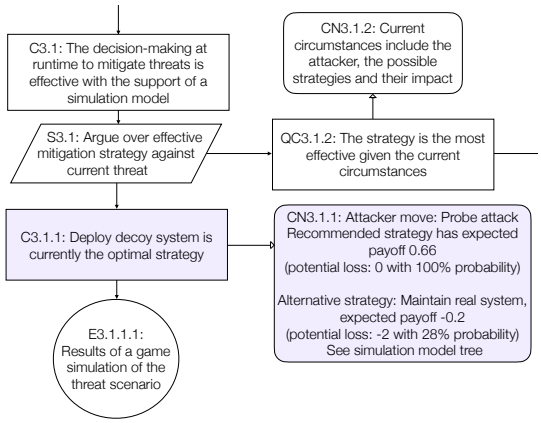

Figure 8: SAC Extension: Threat to be Mitigated

represent a current version of the system's security and what the SAC claims needs to happen to restore or maintain security. Additionally, given that there are usually multiple mitigation strategies available to a decision-maker in this instance, it is more accessible to a human to identify what action to take and be able to make quick decisions with a dynamic claim that shows the current best action. Therefore, C3.1.1. will change to the new C3.1.1. in Figure 8 once an attacker has taken action against the system and the security team would need to react. For each defense strategy that needs to be taken by a security team and therefore for each attack against the asset, claims will be added to represent what actions need to be taken. The evidence connected to these dynamic claims will then be the game-theoretic model that has now considered the new circumstances that impact the system's security.

In addition to the claim, the context node will also change to represent the new circumstances of what strategy the claim proposes. In the context node for the strategy in C3.1.1. the expected payoff and potential loss are described, where the potential loss represents the worst possible payoff under the current circumstances. Moreover, the alternative or next-best strategy with its expected payoff and potential loss is also included to enable quick decision-making and an explanation as to why the suggested strategy is considered most effective. By also providing the alternative strategy, it is easier for the decision-maker to understand how the expected payoffs shown in CN3.1.1. relate to each other and thus enables them to assert their judgment on whether the suggested strategy in the claim is the most suitable one to take. In addition to the payoff and loss values, the user can look at a visual representation of the game simulation for an overview of the different strategies and their payoffs, such as the decision tree in Figure 6. Therefore, this information would be able to support validation at design time as well as the security team in making decisions at runtime and help reduce the potential human error from making assumptions, which aligns with the requirements in Section 4.2.

The interviews revealed that many security teams tend to be based on specific products or versions of products. Therefore, the extension was connected to the product or asset, since this will be the part of the system that the proposed mitigation strategy would be evaluated for and consequently represents the 'defending player' opposite the attacker attacking this asset of the system. In this SAC example (Figure 7), the decision support's top claim C3.1. is added to the financial transactions asset C3. In practice, the asset might be even further decomposed into sub-components, which would make possible attacks and mitigation strategies more detailed and limited in number. The exact placement of the claim C3.1. would therefore depend on the exact system. However, it is important to include the claim at asset level as one of the security controls protecting the corresponding asset to maintain the claim's accessibility and usefulness as a decision support at runtime.

4.4 Evaluation of Game Theory Extension

The SAC extension was evaluated by four security experts. They were shown a simulation of the extension, which was followed up with a questionnaire and four open-ended questions. The results are shown in Table 2. There was one missing value in the dataset. The mean of all available answers is 3.667 with the mode being 4, which generally indicates that the extension is perceived to be mostly effective and increases the confidence in the SAC.

4.4.1 Usability. The extension's ease of understanding was rated at an average of 3.25. Therefore, the extension was somewhat easy to understand, however most participants stated that explanations and clarification of the extension and the included game-theoretic concepts were needed to increase their understanding. The context of the extension, i.e. combining game theory with SACs, was initially unfamiliar for some and therefore led to some confusion. I1 mentioned that high-level definitions of the concepts and use cases would be important to include. I1 also explained that since the simulation tree consists of a large amount of dynamic data, it is important to explain how they change to facilitate its usability.

Table 2: Results of Artifact Evaluation, 1: To No Extent, 3: To Some Extent, 5: To a Great Extent

Question	I1	I2	I3	I4	Mean	Mode	Mdn	SD
Ease of understanding	4	2	4	3	3.25	4	4	0.829
Satisfaction with amount of information	3	4	5	3	3.75	3	4	0.829
Satisfaction with type of information	4	2	3	4	3.25	4	4	0.829
Confidence of decisions	3	2	4	4	3.25	4	4	0.829
Effectiveness for team response and decision-making	4	3	5	4	4	4	4	0.707
Increase in confidence of security at runtime	4	3	4	4	3.75	4	4	0.433
Likelihood of working with model or recommending it	4	n/a	5	5	4.67[1]	5	5	0.471[1]
All					**3.667**	**4**	**4**	**0.861**

[1]Value based on n=3

The type of information in the extension was considered somewhat satisfactory. I3 stated that the user is required to have a certain baseline of knowledge about game theory to interpret the provided information, which they assume most people would not have in this situation and cannot necessarily be expected in the security field. I1 also mentioned the importance of understanding the value of the asset as well as which parts of an asset or in what way it has been compromised in an attack to assess the severity of the issue. In the extension, it was difficult to tell which parts of the asset that the extension connects to are affected, how the asset is affected, and the extent of damage an attacker can do, as well as how this is reflected in the payoff to be able to make decisions with confidence. Based on only the payoff, it was difficult to understand whether it represented a high or low impact. I2 also explained that impact is usually expressed as categories and the use of layered probabilities might be difficult to follow. Moreover, the experts stated that the extension should include more explicit clarification in the SAC on what mitigation strategies are available to the user in an attack scenario. According to I2, including all possible strategies instead of a dynamic claim would give a better overview of the choices that a decision-maker would have.

In addition to the type of information, another measure was based on the satisfaction of the amount of information that was included in the extension. I3 considered the type of information to be very satisfactory after the extension had been explained, where the decision tree would help individuals understand how decisions and the recommended strategy are being generated and what information they are based on with, for example, the weights. I4 also stated that decision trees are a common visualization tool for decision-making processes and therefore would be understandable for most people. However, I1 expressed that more motivation behind how probabilities and payoff values are set or will change

would be needed in order for them to be able to estimate the severity of the issue and understand when these values change due to new incoming information. I2 also mentioned that they would like to know how an attack might degrade the system, which would be encapsulated in the payoff. Consequently, the payoff representing the impact of an attack on the system requires more context and having an aggregate payoff value might not be sufficient.

4.4.2 Effectiveness. Another aspect of the extension was its potential effectiveness in relation to confidence in the decision-making at runtime, how effective it can be for the team's overall decision-making and response as well as how realistic and applicable it is in practice. The extension's effectiveness for security processes such as the overall decision-making and response was rated at a mean of 4 and shows that the extension is considered useful for supporting the activities of a system's security response. The extension was added as a security control to the assets in the system, which I1 considered to be a suitable approach for incorporating the decision support in the SAC. I3 stated that for users who are not directly involved with incident response but with the asset itself and who might not be security professionals, the extension and the decision support it provides could facilitate their participation in, for instance, discussions regarding threat mitigation.

Furthermore, the ability to confidently make decisions at runtime with the extension, which reflects its potential effectiveness and also the perceived trust in the system's recommended strategy, was considered somewhat effective. I4 stated that the structured format of the extension with the probabilities helps give a sense of where an attack might lead. However, I2 found it difficult to answer since in self-adaptive systems the system should be making the decisions that are meant to be supported with the extension proposed in this research. Moreover, concepts such as payoffs might be difficult to gauge for humans. On the other hand, I3 reported that similar types of decision support are already used only under different circumstances and if the proposed extension had an effective game-theoretic model that can deliver recommended strategies in actual attack scenarios it could serve as an effective guide.

Moreover, when faced with an active attack that the security team is able to react to with pre-defined alternatives, I1 stated that this decision support would be realistic since it can evaluate the different options at runtime. I2 expressed doubts regarding the use of a SAC for runtime decision-making, especially when there is little time since SACs could be too complex to navigate and hinder fast incident response. However, they mentioned that it would depend on how incident response teams operate and train for attack situations. I3 considered the extension to be most useful for security decisions on a more systemic level. Moreover, I3 and I4 stated that it would depend on the attack and the attacker whether this extension would be realistic since the worst attacks are by attackers that are well-prepared and subsequently happen very fast without time to react or a human to deploy a defending move.

4.4.3 Impact on Confidence. In regards to how much the confidence in the SAC would increase, most answers rated the extension at a 4 with the mean being 3.75. I4 found that the extension can help consider the runtime aspects of a system and therefore increase the confidence in the case. I1 mentioned that the extension can provide evidence for the continuous monitoring activities at runtime, which would add more validity to the claims regarding the security of the system. Moreover, I3 stated that the extension can help provide more automation to define what processes to follow which can help ensure continuous compliance. I2 stated that the increase in confidence in the SAC would depend on what exactly a system can do on its own without human interference in terms of security. Furthermore, in the evaluation the participants often described their reasoning for the system security in terms of the attacker's point of view, such as keeping the attacker's payoff low reflects more security or considering what the adversary's ideal strategy would be which is what would need to be countered. According to I1, one aspect of threat modeling is to consider how much effort it is for an attacker to attack a part of a system and how much it would be worth to them. However, the focus of the extension, i.e. what is documented in claims and context nodes, was primarily on the system and how the security would be maintained with certain actions. Therefore, the attacker payoff appears to play a more significant role in the decision-making process regarding how to mitigate an attack and should also be included in, for example, the context node.

Finally, when the participants were asked about the likelihood of recommending the tool, most considered the concept to be potentially useful after further development and testing.

4.4.4 Use Cases of the Extension. The participants mentioned multiple areas, where they thought the extension and decision support could help improve different security-related activities. I1 and I2 mentioned the extension's potential usefulness as a validation tool for a system and being able to anticipate how it would work at runtime. They also mentioned that it could be useful in penetration testing, which could serve as evidence in the SACs and thereby increase its confidence. Moreover, I1 and I4 mentioned how the extension could be useful for training people and improving security capabilities of, for instance, the incident response. I3 and I4 also considered the extension to be useful in coordinating and designing procedures around monitoring as well as response plans and escalation processes for when incident response teams need to be brought in since it can bring perspective to how the system would operate at runtime. They stated that many companies do not have a fully operational or sufficient incident response and a decision support would be useful to help guide the security processes on a more systemic level. Additionally, I3 and I4 mentioned that the extension can be used by a broader user group, which might also reduce the reliance on the incident response at runtime. Another aspect where the tool was considered to be potentially useful is in the logging of attacks, which I1 considered to be useful for forensic analyses and creating reports for how to or not to react to certain attacks, which can also be used as evidence to strengthen the SAC.

5 DISCUSSION

This section discusses the findings in light of related work.

5.1 RQ 1: Challenges of SACs at Runtime

Multiple challenges were identified regarding the maintenance of SACs at runtime and being able to use them as effective support for decision-making when faced with a threat. Two of the identified challenges that relate to SACs at design time are the system

and environmental complexities as well as potential restrictions from the system scope. Uncertainty was considered by, for instance, Weyns et al. [29] and De Lemos et al. [7] as the main challenge for SACs at runtime, which then takes place in different forms and through different sources, such as due to incomplete information or the involvement of humans. Many of the challenges that were identified in this research align with the sources identified in the authors' taxonomies, only in this research the focus was on the context of design time and runtime. One aspect that might however not be included in the existing taxonomies are the organizational challenges, specifically in relation to culture and the perceived value of SACs, which impact the management of the uncertainty. While a more effective SAC that can consider actions at runtime could increase the value placed on SACs, the organizational limitations identified in this research are still a broader issue that would involve active participation of stakeholders such as management in companies for a runtime-adapted SAC to be able to take effect and support decision-making at design time and runtime. Moreover, the maintenance of SACs along with the system they represent is essential, especially in more security or safety-critical contexts, to be able to provide assurances of certain system requirements to be met [26]. Therefore, it is central to have effective organizational processes that enable SAC maintenance.

Key Insight: We identified seven challenges for maintaining SACs at runtime. At design time, the difficulty of anticipating uncertainties as well as organizational limitations inhibit SACs. At runtime, the complexity of security incidents and the human aspects obstruct effective SACs.

5.2 RQ 2: Impact of Game Theory-based Decision-making on the Confidence in SACs

Based on the final evaluation, the SAC extension needs to be developed to become more effective and easier to understand. Overall, the participants found it difficult to understand the extension without explanations or context. The idea of making SACs more runtime adaptive by including game theory to support design time and runtime decision-making is a relatively new and complex concept, both for assurance cases within the security as well as safety domain. Moreover, information such as the system impact is needed to have more context on both payoff values and the attack so that humans' evaluations and subsequent decision-making are better supported. However, adding more explanations and information to, for instance, the simulation tree, would still need to be weighed against how much information the extension should contain overall. Since the example game represents a simple version of an attack, where in practice there might be more choices, too much more information can become too complex for a human to understand.

Some participants also preferred a mapping of all available alternatives over a dynamic claim to understand what is possible and weigh the options against each other. The motivation behind having a claim that can dynamically change to reflect the system's state at runtime and show the ideal solution is to help facilitate the use of SACs in situations where fast decision-making is required. The need to be able to follow the decision-making through more context such as by seeing all available strategies relates to the theme of balancing the automated and human aspect of a decision support that was identified in the first iteration. This also involves again a trade-off between the amount of information that is required to make decisions about attack mitigation and maintaining low complexity and high accessibility to the necessary information. Furthermore, the evaluation showed that the extension might not be effective for self-adaptive systems, since humans in this context usually take on a different role, which translates to the SAC, as there is less of a need to manage the system at runtime.

Another limitation of the model in this study is, for instance, the fact that the proposed extension was modeled with a relatively small custom dataset to illustrate the concept in the evaluations with the security experts. It has not been validated with any other datasets. Therefore, in future work the model would also need to be evaluated with different and larger datasets to enable an evaluation of the proposed model in a controlled environment.

Moreover, as mentioned by the participants of this study, game theory can be relatively complex and subsequently difficult to understand especially at runtime when the decision-making is constrained in terms of time. Therefore, in future research it would be important to identify simpler approaches to game-theoretic decision-making in security to make it more usable as well as accessible to different stakeholders.

Key Insight: Three requirements were identified for SACs to enable dynamic decision support and have a positive impact on their confidence. However, the evaluation showed that the game-theoretic information added to SACs needs to be balanced with the increase of complexity in the SAC.

5.3 RQ 3: SAC Extension with Game Theory

The SAC extension is proposed to be added as a security control in connection with the assets in the case and gives strategies to mitigate threats against these assets. Another alternative that was considered was connecting the extension at the attack level in the SAC and connecting it to specific attacks following a threat-based argumentation strategy. While this layout might enable clarity as to what attack the optimal strategies would help mitigate, in practice it is not always clear from the start of a threat what type of attack is taking place. It would be difficult or not possible to correctly connect a mitigation strategy proposed by the decision support to a specific attack. The SAC could also become too complex since the threat landscape is very dynamic.

The implementation of this extension is similar to the SAC adaptation proposed by Jahan et al. [9] for self-adaptive systems in the sense that the system and its environment are continuously monitored to provide the information that is then analyzed by the game-theoretic model to give a suggested mitigation strategy. However, the execution step would be decided on by a security team, instead of only the system. The extension in this paper is expected to be used by humans and therefore includes more information to enable them to make their own risk assessments and increase the confidence in the SAC. Moreover, the proposed extension would support decision-making at runtime and share similarities with

other dynamic risk management frameworks such as the Observe, Orient, Decide, and Act (OODA) framework [3]. However, in our model the focus is primarily on supporting the 'Decide' step.

Apart from the SAC design, the extension is only able to inspire confidence and provide decision support if the game-theoretic model is also effective in suggesting the ideal strategy. In this paper, a relatively simple model was used to simulate the extension, which considers some uncertainties of an attack scenario such as the attacker type. The model's formation and algorithm were not the primary focus of this paper. Instead, emphasis was placed on identifying the requirements that this type of model would need to fulfill to support decision-making. In practice, a more complex model would be needed that is able to consider the uncertainties and impact of available choices to a larger extent. However, the effectiveness of more complex game models will still need to be weighed against their explainability to enable human understanding and confidence in the overall SAC. These concerns about the extension's scalability were also pointed out by the experts in the evaluation when the number of choices for mitigation strategies increases as well as the types of threats that have to be considered.

Additionally, the extension proposed in this study might be too simple to be able to account for different time scales of attacks in interdependent systems. Especially in cyber-physical systems or systems of systems, attacks can propagate at different speeds throughout the system, which would require more complex mitigation strategies. Managing the varying propagation speeds and dynamic interactions in interdependent systems poses a broader challenge in the security domain that needs to be addressed in future research.

Key Insight: The SAC extension was added as a security control. It consists of a claim that dynamically changes to the ideal strategy in a given scenario based on the output of the game-theoretic simulation model.

6 CONCLUSION

In this research, our aim was to investigate how SACs can be extended with game theory to include dynamic decision support at design time and runtime, thereby increasing the confidence in SACs. Since security is a dynamic concept, it is crucial that SACs are able to consider new information at runtime to ensure system security. We extend a SAC with a dynamic claim to reflect runtime threats and what strategies to take to keep the system secure, which is based on a game-theoretic model as the evidence. The extension helps guide the decision-making of security teams making SACs more usable as well as easier to maintain at runtime. In addition to this, it has the potential to serve as a support for validating claims at design time. The evaluation provided insights on potential improvements concerning the extension's scalability and the effectiveness of a SAC for runtime decision-making, particularly during incident response where there is little time, which need to be addressed in future work. This paper presents an initial concept that has to be further developed and tested in an industrial context.

ACKNOWLEDGMENTS

Thank you to the participants of the study for their insights. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] Andy Applebaum, Doug Miller, Blake Strom, Chris Korban, and Ross Wolf. 2016. Intelligent, Automated Red Team Emulation. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*. 363–373.
[2] Erfan Asaadi, Ewen Denney, Jonathan Menzies, Ganesh J Pai, and Dimo Petroff. 2020. Dynamic Assurance Cases: A Pathway to Trusted Autonomy. *Computer* 53, 12 (2020), 35–46.
[3] John R. Boyd et al. 2018. *A Discourse on Winning and Losing*. Vol. 400. Air University Press.
[4] Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101.
[5] Radu Calinescu, Danny Weyns, Simos Gerasimou, Muhammad Usman Iftikhar, Ibrahim Habli, and Tim Kelly. 2017. Engineering Trustworthy Self-adaptive Software with Dynamic Assurance Cases. *IEEE Transactions on Software Engineering* 44, 11 (2017), 1039–1069.
[6] Anna-Karin Carstensen and Jonte Bernhard. 2019. Design Science Research – A Powerful Tool for Improving Methods in Engineering Education Research. *European Journal of Engineering Education* 44, 1-2 (2019), 85–102.
[7] Rogério De Lemos, David Garlan, Carlo Ghezzi, Holger Giese, Jesper Andersson, Marin Litoiu, Bradley Schmerl, Danny Weyns, Luciano Baresi, Nelly Bencomo, et al. 2017. Software Engineering for Self-Adaptive Systems: Research Challenges in the Provision of Assurances. In *Software Engineering for Self-Adaptive Systems III. Assurances: International Seminar, Dagstuhl Castle, Germany, December 15-19, 2013, Revised Selected and Invited Papers*. Springer, 3–30.
[8] International Organization for Standardization and Society of Automotive Engineers. 2021. ISO / SAE 21434 Road vehicles – Cybersecurity Engineering. Retrieved February 20, 2024 from https://www.iso.org/standard/70918.html
[9] Sharmin Jahan, Ian Riley, Charles Walter, Rose F Gamble, Matt Pasco, Philip K McKinley, and Betty HC Cheng. 2020. MAPE-K/MAPE-SAC: An Interaction Framework for Adaptive Systems with Security Assurance Cases. *Future Generation Computer Systems* 109 (2020), 197–209.
[10] Charles A Kamhoua, Christopher D Kiekintveld, Fei Fang, and Quanyan Zhu. 2021. *Game Theory and Machine Learning for Cyber Security*. John Wiley & Sons.
[11] Kevin Leyton-Brown and Yoav Shoham. 2008. *Essentials of Game Theory: A Concise Multidisciplinary Introduction*. Springer Nature.
[12] Xiannuan Liang and Yang Xiao. 2012. Game Theory for Network Security. *IEEE Communications Surveys & Tutorials* 15, 1 (2012), 472–486.
[13] Elena Lisova and Aida Čaušević. 2018. Towards Security Case Run-Time Adaptation by System Decomposition into Services. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 4102–4108.
[14] Kong-Wei Lye and Jeannette M Wing. 2005. Game Strategies in Network Security. *International Journal of Information Security* 4, 1 (2005), 71–86.
[15] Nikolai Mansourov and Djenana Campara. 2011. *System Assurance: Beyond Detecting Vulnerabilities*. Elsevier.
[16] John McInerney, Stephen Stubberud, Saquib Anwar, and Stephen Hamilton. 2001. FRIARS: A Feedback Control System for Information Assurance Using a Markov Decision Process. In *Proceedings IEEE 35th Annual 2001 International Carnahan Conference on Security Technology*. IEEE, 223–228.
[17] Mazen Mohamad, Örjan Askerdal, Rodi Jolak, Jan-Philipp Steghöfer, and Riccardo Scandariato. 2021. Asset-driven Security Assurance Cases with Built-in Quality Assurance. In *2021 IEEE/ACM 2nd International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS)*. IEEE, 1–8.
[18] Mazen Mohamad, Alexander Åström, Örjan Askerdal, Jörgen Borg, and Riccardo Scandariato. 2020. Security Assurance Cases for Road Vehicles: An Industry Perspective. In *Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES20, 29)*. Association for Computing Machinery, 1–6.
[19] Mazen Mohamad, Rodi Jolak, Örjan Askerdal, Jan-Philipp Steghöfer, and Riccardo Scandariato. 2023. CASCADE: An Asset-driven Approach to Build Security Assurance Cases for Automotive Systems. *ACM Transactions on Cyber-Physical Systems* 7, 1 (2023), 1–26.
[20] Mazen Mohamad, Jan-Philipp Steghöfer, and Riccardo Scandariato. 2021. Security Assurance Cases - State of the Art of an Emerging Approach. *Empirical Software Engineering* 26, 70 (2021), 472–486.
[21] Omar Ochoa, Jessica Steinmann, and Yevgeniy Lischuk. 2018. Towards Eliciting and Analyzing Security Requirements using Ontologies through Use Case Scenarios. In *International Conference on Software Security and Assurance (ICSSA)*. IEEE, 1–6.
[22] Annapurna P. Patil, Bharath S, and Nagashree M. Annigeri. 2018. Applications of Game Theory for Cyber Security System: A Survey. *International Journal of*

Applied Engineering Research 13, 17 (2018), 12987–12990.

[23] John Rushby. 2008. Runtime Certification. In *International Workshop on Runtime Verification: RV08*. Springer, 21–35.

[24] John Rushby. 2015. *The Interpretation and Evaluation of Assurance Cases.* Tech. Rep. SRI-CSL-15-01. Comp. Science Laboratory, SRI International.

[25] Yuanzhuo Wang, Chuang Lin, Kun Meng, and Junjie Lv. 2009. Analysis of Attack Actions for E-Commerce Based on Stochastic Game Nets Model. *Journal of Computers* 4, 6 (2009), 461–468.

[26] Charles B Weinstock and Howard F Lipson. 2013. *Evidence of Assurance: Laying the Foundation for a Credible Security Case.* Software Engineering Institute Report. Carnegie Mellon University.

[27] Antonia Welzel. 2024. Evaluation Guide Iteration 2. https://figshare.com/articles/journal_contribution/Evaluation_Guide_Iteration_2/25225766

[28] Antonia Welzel. 2024. Interview Guide Iteration 1. https://figshare.com/articles/journal_contribution/Interview_Guide_Iteration_1/25225760

[29] Danny Weyns, Nelly Bencomo, Radu Calinescu, Javier Cámara, Carlo Ghezzi, Vincenzo Grassi, Lars Grunske, Paola Inverardi, Jean-Marc Jezequel, Sam Malek, et al. 2017. Perpetual Assurances for Self-Adaptive Systems. In *Software Engineering for Self-Adaptive Systems III. Assurances: International Seminar, Dagstuhl Castle, Germany, December 15-19, 2013, Revised Selected and Invited Papers*. Springer, 31–63.

[30] Jianjun Zheng and Akbar Siami Namin. 2018. Defending SDN-based IOT Networks Against DDOS Attacks Using Markov Decision Process. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 4589–4592.