CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Increasing the confidence in security assurance cases at runtime

A decision-support using game theory

Master's thesis in Computer science and engineering

Antonia Welzel

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

Increasing the confidence in security assurance cases at runtime

A decision-support using game theory

Antonia Welzel

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

Increasing the confidence in security assurance cases at runtime
A decision-support using game theory
Antonia Welzel

Supervisor: Mazen Mohamad and Rebekka Wohlrab, Department of Computer Science and Engineering
Examiner: Birgit Penzenstadler, Department of Computer Science and Engineering

Master's Thesis 2023
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2023

Increasing the confidence in security assurance cases at runtime
A decision-support using game theory
Antonia Welzel
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Security assurance cases consist of arguments which are supported by evidence to justify that a system is acceptably secure. However, security assurance cases are relatively static and therefore currently not effective at runtime in supporting users to mitigate threats. The aim of this thesis was to investigate how security assurance cases can be extended with game theory in order to enable dynamic decision-support in the context of threats and environmental changes. Game theory is able to represent the interaction between different actors and identify their optimal strategies based on their payoffs and likelihoods.

In order to identify the relevant requirements for a security assurance case extension, interviews were conducted with security experts to identify what challenges there are with maintaining security assurance cases at runtime that make them not able to effectively support decisions. The security assurance case extension was then created based on these findings and in the end evaluated with the security experts in order to assess its effectiveness.

The results show that there are multiple challenges both at runtime itself as well as design time towards maintaining security assurance cases and enabling them to become a more 'living' document. Some of the challenges were, for instance, uncertainty due to the system and environmental complexity, organizational limitations such as ineffective maintenance processes as well as complex decision processes at runtime. Moreover, an effective decision-support as part of security assurance cases would need to be able to simulate decision-making at runtime to guide the strategy in attack scenarios with humans in the loop in order to subsequently manage the different challenges.

The extension of the security assurance case was added as a security control connected to assets in the security assurance case, where a 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. The concept of integrating more runtime adaptivity is new and relatively complex. Overall, based on the results of the evaluation, the extension was considered as being potentially useful, however this would further depend on how it will be implemented in practice.

Keywords: Security Assurance, Security Assurance Cases, Game Theory, Dynamic Decision-making, Runtime.

Acknowledgements

I would like to thank my supervisors Rebekka Wohlrab and Mazen Mohamad for all their valuable feedback, guidance and help. I would also like to thank my examiner Birgit Penzenstadler for her feedback. Moreover, thank you to all the security experts for participating in the interviews and evaluations of the results and for providing useful insights and feedback. Finally, I also want to thank my family and friends for their support.

Antonia Welzel, Gothenburg, June 2023

Contents

List of Figures

List of Tables

1
Introduction

The chapter presents the research topic, its background and the purpose of this study.

1.1 Security Assurance of Systems

The security of a system has become increasingly important with higher levels of connectivity and the subsequent more extreme consequences of cyberattacks [1]. Therefore, providing assurances for the security of a system is also becoming a more essential part of system development and maintenance. One way to assess and subsequently ensure security in a system are security assurance cases, which can be used to increase confidence in the security of the system as well as identify potential weaknesses [2]. Security assurance cases consist of a set of arguments and evidence which build a case describing why the system can be considered acceptably secure. For each argument in the assurance case, there is the overall claim that a system is acceptably secure, which then branches into different sub-claims that ultimately end up in a final analysis in the form of evidence to show that they are true [3].

Many studies have so far focused on applying security assurance cases in different contexts and industries, however there is not yet much research focusing on how the security of a system can be ensured aside from assumptions and requirements that support the claims and arguments. Moreover, there is still little research in regards to how security assurance cases can be maintained at runtime and support more continuous maintenance as well as compliance. Additionally, security assurance cases show a more static version of decision-making and evidence of the system's security at design time. However, when an attack occurs, due to for example unexpected behaviour, the security claims in the assurance cases might not hold and therefore consideration of how the system behaves at runtime is needed.

1.2 Problem Statement

Security assurance cases are used to document the system's quality specifically in terms of security and additionally with this documentation the system's compliance with certain standards and regulations can be checked and maintained. Therefore, security assurance cases represent an important design document to show that the system is acceptably secure, which is generally difficult to define since it is very context-dependent as well as complex. As a result, the assurance cases can then

help increase the confidence in the system's security, which in this context is defined as the belief that the system's security is reliable.

However, security is dynamic, which means there will be changes to the system at runtime. These changes have to also be reflected in the security assurance cases in order to maintain their claims and evidence, and consequently the confidence they provide for the system in question. Hence, security assurance cases would need to provide some form of reassurance that the system at runtime is still in compliance with the standards they should have, which would then also maintain the confidence in its quality. Therefore, as the relevance of a security assurance case is dependent on how it is maintained and able to reflect the current state of the system, it would then be necessary to be able to understand how possible attacks would affect the system or cause it to react in order to support the design and decision-making processes.

One way to approach this issue could be through game theory. Game theory studies the interactions between independent self-interested agents, where the actions of each actor is modeled to understand how they will affect each other and what the outcomes might be [4]. Different possible plays of action can be modeled to identify the outcome. In the context of system security this would be through for instance an attack-defense model. Currently, the assessment of the security of the system is based on the evidence, assumptions and arguments in the assurance cases. Using game theory, the level of security of a system could be modeled based on the arguments and evidence in assurance cases as a sequence of 'moves' made by the system and how secure they are from an attack while also considering the uncertainties of unknown events that might occur once the system is in use.

While there have been many connections between game theory and computer or network security, where games have been used to for instance model different attack or defense strategies, there is still a gap in the literature regarding game theory and security assurance cases. Considering that industry standards and regulations are continuously changing, the maintenance of security assurance cases is very demanding and therefore they would benefit from incorporating a process for continuous compliance, such as runtime adaptation using game theory to keep up with the changes in their domain. Therefore, this issue would be relevant for both research purposes as well as in practice.

1.3 Aim

The purpose of this thesis is to explore challenges with maintaining security assurance cases at runtime, and to investigate how security assurance cases can be extended with game theory to be able to consider the system's behaviour at runtime by creating a decision-support mechanism for designing security assurance cases and maintaining them as their environment changes. By including game theory, the decision-making process as well as the maintenance can become more dynamic to increase the confidence in the security assurance case and also in turn the system's security.

1.4 Research Questions

In order to improve the design and maintenance of security assurance cases and approach a possible solution towards security assurance cases becoming more of a 'living' document, a better understanding of the main issues with security assurance cases and how they are not currently effective at runtime is required. Therefore, the first research question in this thesis is the following,
RQ 1: What are the current challenges with maintaining the security assurance cases at runtime?

Moreover, security assurance cases increase the confidence in the system since they represent and provide a form of reassurance that it is in compliance with the industry standards and regulations and therefore also proves the level of security of their system. However, it is important to understand how much of an effect game theory as decision-support has on increasing the confidence in the case and whether more dynamic decision-making can improve security assurance cases' effectiveness at runtime and therefore the confidence in them. Consequently, the second research question is,
RQ 2: To what extent can the confidence in security assurance cases be increased with game theory based decision-making?

The last research question focuses on how security assurance cases are extended with game theory to be able to evaluate multiple strategies and consider the uncertainties in the environment that take place at runtime.
RQ 3: How can security assurance cases be extended to include runtime behaviour using a model based on game theory?

1.5 Disposition

The next chapter describes the theoretical background to security assurance cases as well as game theory. Following this, the related work within cybersecurity and security assurance cases at runtime is presented. The chapter after describes the method of the project and how the research questions were answered. Furthermore, the results are presented and discussed. Lastly, limitations and possible topics for future research are discussed.

2
Theoretical Background

In the following section, the theoretical context to security assurance cases and how they provide confidence in the security of a system is presented. Moreover, the section includes background on game theory and typical algorithms that are also often related to cybersecurity.

2.1 Security Assurance Cases

Security assurance cases are sets of claims and arguments for which evidence is provided to give information and proof about a system's cybersecurity. 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 [2]. For each claim, there is a set of arguments to show that the claim holds for the system with support of the arguments and related evidence [5]. This then increases the confidence in the system that it is secure or "acceptably secure", as it is typically described in security assurance cases [6], which refers to the related claims and evidence satisfying their security requirements and providing sufficient confidence in the system security based on the current information and context.

Security assurance cases can be expressed either in textual or graphical form and there are different approaches for creating them such as the CASCADE framework [2]. The most common form of documentation of the security assurance cases for a system is performed with Goal Structuring Notation (GSN) or Claims, Arguments and Evidence notation (CAE) [2]. An example of a security assurance case is shown in Figure 2.1. Claims in security assurance cases are statements about a property of the system that is then argued for and evidence provides proof for it. Additionally, context nodes are usually connected to claims in order to give the context to what is stated in the claim and when it applies [2]. Mohamad et al. [2] also consider 'case quality' claims as another form of claim that relate to the quality of the components used in the security assurance case to justify the system's security such as other claims or evidence.

The arguments in the security assurance cases then argue about specific properties of the system for why the claim holds. As security assurance cases have one top claim, all claims together with their arguments and the evidence that follow it make up the arguments for this top claim. The cases' argument structure can follow different approaches, such as being based on assets, security requirements, threats or

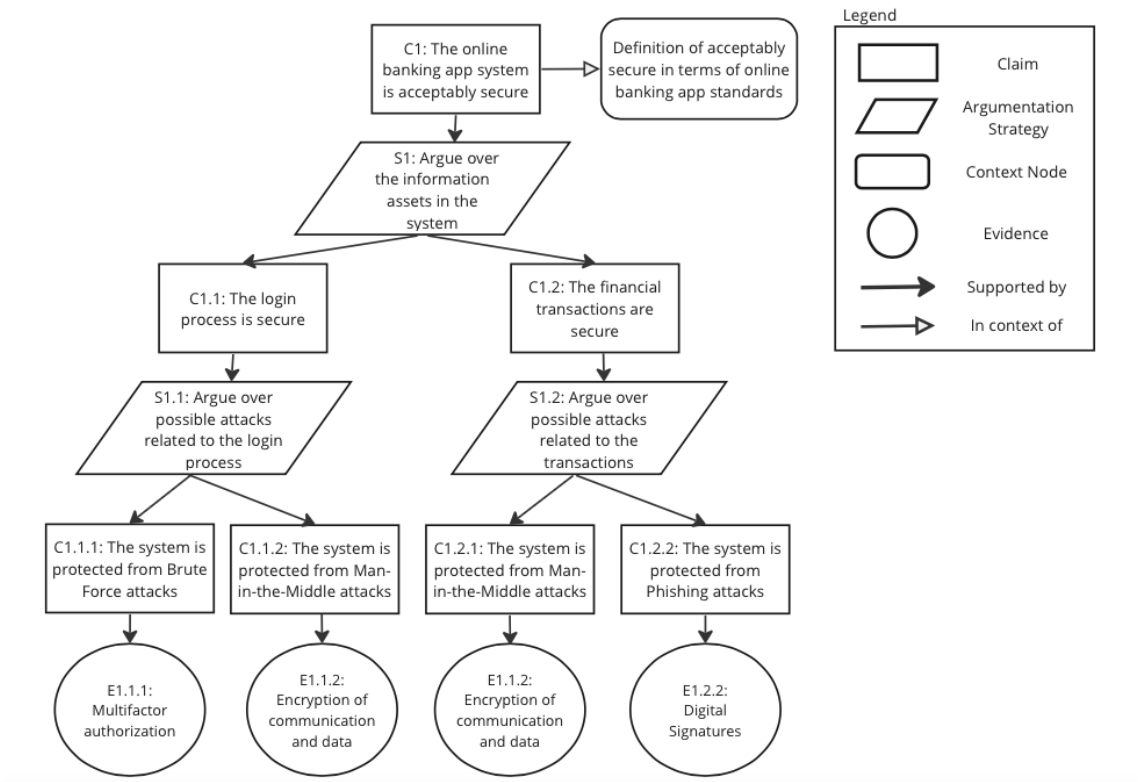

Figure 2.1: Example of Security Assurance Case

attack paths, which specify what system properties are in focus for the set of arguments [2]. Therefore, the argumentation strategy needs to be set to define what part of the system will be argued for and what to structure the claims and subsequent arguments after.

Lastly, the evidence is presented at the end of a security assurance case to support the claims and arguments. It can relate to either the technology or to the processes maintained by people to ensure the security of a system. Evidence needs to inspire confidence in the previously made claims and it is therefore important to choose a sufficiently credible evidence type [7].

2.2 Quality Assurance for Security Assurance Cases

In order to provide sufficient assurance for a system's security, effective processes have to be set in place along with the required technology as well as the right people [6]. Moreover, the usefulness of security assurance cases and the confidence they provide for the system's security is dependent on whether they are of sufficient quality, such as effective argumentation structures and the right evidence [2]. Evidence in these cases is according to Lipson and Weinstock [7] typically the most impactful factor towards providing confidence in the security assurance case's validity and credibility of its claims. If the provided evidence is too weak and not fit for the context of the claims then these claims and the case overall will not be credible.

Security claims typically relate to either the security requirements for the system or how the system is not vulnerable to potential attacks, where then vulnerabilities might arise when the claims rely on false or incomplete assumptions which do not apply in reality [7]. Moreover, the cases are typically created at design time [6]. This means that once the cases have been constructed there might be errors or issues presented at runtime that were not considered in the design process. Therefore, a common challenging aspect to security assurance cases is the lack of runtime considerations. Security domains are exposed to high uncertainty, especially compared to safety domains, since there is often insufficient knowledge about potential attacks and attackers' abilities, which makes it more difficult to anticipate the necessary or correct mitigation strategies [8]. In more dynamic environments, such as security domains or more adaptive systems, there is consequently an increased risk of security threats taking place at runtime that were not anticipated and therefore covered in the security assurance case. This would then increase the risk of harm to the system and also decrease its security assurance and the confidence in the claims made about the system's security since they might not apply.

Subsequently, more adaptivity in security assurance cases to external circumstances would be required to be able to keep up with external changes taking place that might impact the system. A system, especially self-adaptive or protecting one, needs to be able to either react with a mitigation strategy or proactively anticipate threats at runtime [9]. However, when security assurance is developed at design time the result is often too static to provide enough confidence also at runtime [6]. At the same time, it is challenging to develop dynamic security assurance while still maintaining compliance with existing security requirements both at design and runtime.

2.2.1 Maintenance of Security Assurance Cases

Security assurance cases help document and therefore also identify problem areas in a system in terms of its security. If changes occur to the system, then security assurance cases can be used to understand if there is a potentially new threat that needs to be mitigated. However, creating security assurance cases is a difficult and expensive task due to the large number of security requirements that companies have to typically adhere to and moreover the requirements' interpretability further increases the difficulties in regards to their satisfaction [10]. The complexity of the system and therefore the evidence that is created can become a challenge to apply in security assurance cases [7]. In addition, the maintenance process of security assurance cases is also relatively time-consuming, since security is dynamic and therefore security requirements are continuously changing which needs to be reflected by security assurance cases. Lipson and Weinstock [7] also mention the importance of implementing version control and the traceability to security requirements for the evidence that is provided in assurance cases in order to increase their validity.

In order to ensure that security assurance cases provide a high level of confidence,

the evidence is updated or extended with information collected at runtime [11]. Runtime verification is a form of testing of the system at runtime and verifying that the assumptions are actually true. This will then provide evidence for the system's behaviour at runtime and help increase the assurance for claims that correspond with the evidence. It is therefore an important part of the maintenance process of security assurance cases in order to validate that there are no new issues and the system is in compliance with its security properties.

However, this verification often takes place after the system is deployed or in a test environment with conditions similar to deployment and therefore does not enable any knowledge of the system at runtime beforehand. Runtime certification on the other hand is a form of proactive anticipation of runtime behaviour of a system, where assumptions in assurance cases are monitored to predict possible issues. For instance, Rushby [12] uses Bayesian Belief Networks to represent the security assurance cases and give an estimate on the confidence of an assumption made at design time and how secure it can be assumed to be at runtime [11]. These types of estimates and measures also align with the evaluation methods mentioned in Section 2.2.2.

2.2.2 Evaluating Security Assurance Cases

In order to gain a better understanding of how reliable and credible a security assurance case is for the system it represents, there are multiple measures used for evaluations of security assurance cases. Therefore, these evaluations can help increase the confidence in the security assurance case's claims. A form of this is proposed by Lin, Shen, and Cheng [13], where the authors calculate confidence levels for claims in security assurance cases based on existing similar cases that are proven to be valid.

Another approach is to base the evaluation on the evidence and to move bottom-up in the security assurance case. The existing research of the assessment of security assurance cases and how strong their evidence is, is described by Nair et al. [14] where they perform a systematic literature review of different papers and the techniques they describe. The techniques were sorted into qualitative and quantitative assessments, checklists and logic-based assessments. Quantitative measures come in the form of, for example, the Modus approach where quantitative values based on expert estimations and probabilistic simulations are combined with arguments. Moreover, Bayesian network evaluations are also used as a quantitative measure to estimate the confidence in the overall security assurance case. However, this also requires security assurance cases to be represented as directed acyclic graphs or Bayesian Belief Networks in order for this type of evaluation to be possible.

Overall, many forms of evaluations of security assurance cases rely on more static measures, which means they are not able to consider what might happen later in the future or once the system is in use. This reduces their ability to provide confidence in the assurance that the security assurance cases provide and therefore in the system's security [15]. Pham et al. [16] developed a system for assessing security

assurance in a more continuous form with multi-agents attack graphs. It consists of both an "attackability" metric that statically evaluates a system and anomaly detection based metrics that are used for more dynamic evaluations at runtime. According to the authors, the basis for security assurance assessments should be on the factors that affect the security assurance of a system or the consequences of a certain level of security assurance. From a static view, both these aspects can be measured, whereas at runtime only the factors affecting a system can be accurately measured since the consequences have not happened yet [16]. Therefore, considering both static and dynamic values are important for gaining an accurate understanding of a system's level of security assurance.

The evaluation of security assurance cases can be qualitative or quantitative as described by the different measures that have been mentioned, but they still rely on human judgment in the process. Rushby [12] points out that people can be susceptible to confirmation bias, where their focus is on confirming their hypothesis rather than finding reasons for why it might not hold. Therefore, automated decision-support could help reduce the possible bias in security assurance cases. Additionally, it is important to challenge the claims made and look at what could potentially cause it to not be true anymore.

2.3 Challenges for Security Assurance Cases at Runtime

There are multiple challenges in relation to providing security assurance over a system's lifecycle, such as the uncertainty from insufficient information as well as the uncertainty from humans involved with the system as for instance in the case of self-adaptive systems [17]. According to Weyns et al. [17], uncertainty is the main challenge to security assurance cases overall, in particular for self-adaptive systems, as well as for enabling continuous security assurance. New evidence and arguments have to continuously be collected and included in the cases in order to have reliable assurances when changes occur either in the system or its environment. Moreover, a system that is deployed in environments of uncertainty also affects the type of evidence that can and needs to be collected in order to create a valid case [18].

Perez-Palacin and Mirandola [19] consider uncertainty to arise from two types of sources 'location' and 'nature'. The location relates to the system itself being considered a type of source of uncertainty due to either its overall structure, the input parameters or the system's context. Moreover, nature represents the uncertainties that arise due to either insufficient information about processes or the variability of the environment.

In addition to this taxonomy, Weyns et al. [17] identify four groups of sources of uncertainties that relate to either the system, the system goals, the environment or context of the system and lastly the human aspects. System-related uncertainties are based on the design and how it might be incomplete, simplified, decentralized or

changing due to, for instance, its adaptivity, which will affect the assumptions made and subsequently the assurance case. Furthermore, uncertainty in relation to the system goals refers to accurate requirements elicitation and management in order to define the scope and goals of a system. Additionally, the environment of the system presents another source of uncertainty, which can also take place due to the context of the system being uncertain and evolving over time which needs to be taken into account as well. Finally, the uncertainties originating from human aspects arise due to humans being uncertain themselves which also contradicts the assumption of humans performing tasks correctly. Another uncertainty related to this aspect is the involvement of different stakeholders and multiple ownership, where for example unnotified changes in a third-party application can cause issues in maintaining security assurance cases [17].

Moreover, in many situations security assurance cases need to become automated and possibly also adaptive in order to reflect runtime features in an assurance case. However, there are still challenges in relation to developing approaches and tools that can create and maintain assurance cases with adequate validity [18].

2.4 System Security Management

The process of managing the security of a system both at design and runtime consists of different steps, which are then documented in security assurance cases to ensure confidence in the system security. Nespoli et al. [20] conceptualizes the cyberdefense into four main components; prevention, detection, reaction and forensics.

These constituents affect each other through their feedback. The prevention phase accounts for the monitoring activities in order to prevent attacks such as through firewalls and intrusion prevention systems [20]. Moreover, in order to anticipate potential threats and security concerns at runtime, threat modelling and risk assessments are conducted. Furthermore, since most systems cannot be assumed to be completely secure, intrusion detection systems need to be in place as well. This makes up the detection component. Furthermore, the reaction phase is initiated once an attack is detected, where the impact of the attack is evaluated. In the event of an attack, there is typically an incident response process to assess and mitigate the attack. The aim is to provide mitigation strategies to counter the attack, as well as restore the security of the system [20]. Lastly, forensics of the attack are collected and analyzed in order to for example understand what went wrong and what system parts might need to be updated in order to avoid the incident in the future. These steps in security management are also very similar to how self-adaptive systems maintain security in their systems. Jahan et al. [6] suggest that self-protecting systems should follow a Monitor, Analyze, Plan and Execute with Knowledge (MAPE-K) control loop in order to handle incoming threats.

In the reaction phase or incident response, security analysts have to make complex decisions based on both past as well as current knowledge since security is very dynamic and new information needs to be taken into account [21]. Moreover, multi-

ple components need to be considered such as the complexity of the environment and the attacker in order to make effective decisions. Therefore, aside from technical or procedural security controls, due to the potential human error in decision-making, such as due to biases or lack of capabilities, the human aspect needs to be considered in cybersecurity in order to achieve and maintain the highest possible security of a system [22].

2.5 Game Theory

In this thesis, the aim is to explore the use of game theory for security assurance cases. Game theory studies the interaction between two independent agents. The main components in games are the players, which are the actors involved in the game and affect the sequence of actions that takes place, the actions, which are the moves of the players and can be assumed to be either known or unknown to them, the payoff, which is the return for each player in the game based on their actions, and finally the strategies which are the players' plan of moves in the game for how they will try to win [23].

Depending on the players' strategies and their motivations, there are different types of games in game theory based on the game's different characteristics. For instance, perfect information games are games, where all players know about the possible moves that exist as well as what moves have already taken place, such as in normal-form games. On the other hand, in games with imperfect information, players are not able to observe each other's actions [23]. Moreover, games are also 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 [4]. Additionally, another aspects is whether a game is static or dynamic, where dynamic games are based on multiple moves made by the player. Static games, on the other hand, only involve one move made by each player which ends the game.

Furthermore, among the actions available to a player, a pure strategy means choosing one of the available actions [24]. On the other hand, a mixed strategy reflects one player's uncertainty over the other player's move in reaction to their move. Therefore, the mixed strategy extends the pure strategies with probability distributions and consequently includes more randomness in the action set, since this means the players would randomly choose between the different actions [24]. The best strategy is in the end the one that returns the highest payoff and can be found by identifying an equilibrium of the players' payoffs and strategies [23]. One example of an equilibrium is the Nash equilibrium.

2.5.1 Types of Games

As discussed above, there are different types of games depending on the players, their strategies and the general circumstances of the game. One type of game is is a zero-sum game. In these games, the players have opposing interests, where each

player's win results in the loss and hence worst-case scenario for the other player. The equilibrium in these games can be solved through for instance linear programming. Moreover, another type of game is the Bayesian game. Bayesian games assume incomplete information of the players, where at least one player will not have complete knowledge about the other player's payoffs. In order to account for the unknown information, random variables also referred to as 'types' are included in the game [24]. These describe the beliefs or prior assumptions made about a player. In a game tree, these are represented as a move made by 'Nature' or 'Chance'.

Furthermore, there are stochastic games, which include different 'states' that are transited according to a certain probability to model a dynamic environment [23]. Stochastic games can also be referred to as Markovian games, which are based on Markov chains [24]. Markov chains are mathematical models that illustrate the probabilities of a sequence of events, where each probability is dependent on the state of the previous event [25]. It therefore models the system's behaviour over time and assumes that the current state contains all the information for making predictions on the next states. The transition between states in Markovian games is dependent on the current state of the game and the actions of the players, which determine the transition probabilities for the next state [26]. Hence, Markovian games are able to model dynamic and more complex environments. However, the games themselves can become relatively complex, especially when there are more players, actions and states involved, which can also complicate their interpretation and subsequently the decision-making [27].

Stochastic games can be seen as generalized Markov decision processes since these games extend them by being able to also consider multiple players and their interactions [24]. Markov decision processes are a type of Markov chain, which consist of a sequence of events, where the probability of each event is dependent on the current state of the model. It also includes the decision-making aspect and therefore focuses on finding the best policy to maximize outcomes. Markov decision processes are typically used in combination with perfect information games, however in games with imperfect information, other variations of it have to be used such as partially observable Markov decision processes [28]. Partially observable Markov decision processes are able to model decision-making in contexts, where there is more uncertainty especially about the outcomes of certain actions and therefore finding optimal strategies under these circumstances [29]. The process relies on partial observations of the current state based on which the state can be estimated [30].

However, Markov decision processes are in themselves not considered to be a pure representation of game theory, since the decision process is created by one decision-maker, where they set the possible actions as well as states and determine the transition probabilities and reward functions with the observed information. Therefore, the decisions that are made do not affect the information that is observed and subsequently do not affect the transition probabilities and potential rewards. They are independent of the decision-maker's actions. Decisions within game theory are not as centralized and the interactions between decision-makers or players need to

be accounted for [31]. Therefore in this context, the transition probabilities and reward functions for each Markov decision process would need to be influenced by all players' decision processes in order to give realistic results.

3
Related Work

This section describes research related to security assurance cases with runtime considerations, such as in self-adaptive systems, as well as existing dynamic decision-support for security systems to for instance aid in the mitigation of attacks.

3.1 Security Assurance at Runtime

There are still few forms of evidence that provide security assurance at runtime, which can either give users a false sense of security or reduce confidence in the system [15]. Additionally, the risk assessments conducted during design time before deployment are not always sufficient in giving assurances for the system's behaviour at runtime. The support for runtime adaptation in security assurance cases would help increase the confidence in the system's security, since engineers would be able to understand how the security of their system would play out in reality when it is subjected to different external factors that cannot be accounted for at design time. This would for instance be very useful in the automotive context, where it is often difficult to provide necessary maintenance to a system after a product has been released, such as with cars where software patches cannot always be supplied over air [32].

3.1.1 Security Assurance Cases for Self-adaptive Systems

A type of system where security assurance cases have to be more dynamic to consider runtime behaviour is self-adaptive systems. Self-adaptive systems operate in a relatively dynamic environment, where the system's functionality changes at runtime to work effectively with the new information it receives. Therefore, these systems have to adapt to many uncertainties due to changes within the system or in the system's environment as well as possible security concerns [6]. However, in self-adaptive systems, the implementation of assurance cases that can provide confidence in the security of the system's different states has been difficult. In these systems, many tasks and behaviours are not set and usually unknown until runtime, since they respond to changes in their environment. Therefore, it is difficult to confidently demonstrate with security assurance cases that all requirements are accurately covered [11]. In order to be able to maintain confidence in the system's security, a more dynamic approach is needed to provide assurance such as being able to anticipate threats and the system's behaviour at runtime [9]. Subsequently, security assurance cases also have to be dynamic and consider all possible functionalities that may form

to maintain assurance of the system's security.

As discussed above, according to Jahan et al. [6], self-protecting systems should follow a MAPE-K control loop in order to be able to mitigate threats. This loop is also mirrored by the adaptation of the security assurance cases, which then enables the system to be assessed for how secure it is and provide assurances. When a functionality is changed at runtime, the corresponding security assurance case will then be adapted to reflect these changes. Jahan et al. [6] use security assurance cases with the GSN notation, where changes 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 therefore 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 in the system [6]. Additionally, the main security goals are interdependent and therefore a degrading change in the achievement weights in one goal will also have an effect on other related goals represented in the system.

However, it is important to note that in self-adaptive systems, a key feature is adapting functionality dynamically to uncertainties that arise at runtime. With these adaptations, new vulnerabilities can also arise in the system and therefore the mechanisms for adaptation need to be able to account for these potential issues as well [33]. Moreover, the adaptation operates at runtime, where it is dependent on what actions the self-adaptive system will actually take. Therefore, it is also not a system that anticipates the behaviour at design time but rather being dynamic and adapting at runtime, as opposed to the extension that is proposed in this thesis, which is intended 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. However, a system's capabilities in relation to security need to be overall relatively dynamic, since they are dependent on and typically taking place in response to an attacker's behaviour. Therefore, even though the focus in this thesis is not exclusively on self-adaptive systems, how the dynamic behaviour of self-adaptive systems is accounted for in security assurance cases is a relevant area to consider.

3.2 Dynamic Decision-making in Cybersecurity

Given that cybersecurity is dynamic and is influenced by a changing environment, its ability to consider real-time changes is becoming increasingly important. There are multiple studies within applying more dynamic decision-making in cybersecurity and evaluating risks associated with different threats and mitigations. Risk analysis is an important feature in cybersecurity and is also a function of interactions between players illustrated by game theory [34].

Game theory has been applied before in the network security context specifically for attack-defense models, for instance Lye and Wing [35] used game theory to

model interactions between an attacker and a system, which is based on a multi-player model and aims to find equilibrium strategies. These security games can be used to model attacks and the defense of a system, and 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. This makes these games effective for both defense analysis and performing security assessments of a system [36]. However, security game simulations can be implemented with different types of models depending on the system and the type of interaction between the parties, where 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 [37].

The Markovian decision process is a stochastic algorithm that also has been used at multiple instances to simulate and automate decision-making regarding cybersecurity and defense strategies. McInerney et al. [38] use a Markovian decision process in a single-player game to model a system's defense against attacks. Furthermore, Zheng and Namin [39] 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.

Additionally, Monte Carlo simulations have also been implemented 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 [40].

4
Method

This chapter describes the research method for identifying the challenges for security assurance cases being able to maintain confidence in their system at runtime and how they can be extended to include more dynamic decision-making. A design science research approach was applied where the process was split into two iterations. The first one focused on the challenges of security assurance at runtime which is directly related to the first research question. The second iteration focused then on extending security assurance cases to include more dynamic decision-making based on game theory, which relates to question two and three.

4.1 Research Design

A design science research approach [41] was taken to investigate how security assurance cases should be extended to include dynamic decision mechanisms as well as creating a model of this to in the end be able to answer the research questions. It was considered to be best suited for this project due to the iterative nature of the process. Design science research focuses on the design process and gaining knowledge about designing the artifact as well as the artifact itself [41]. The artifact in this research was developed over two iterations. At the end of each iteration, the artifact was evaluated and validated which served as the basis for the implementations in the next iteration.

The artifact that was developed is an extension to security assurance cases to include considerations of their runtime behaviour and enable continuous compliance. In the first iteration, challenges to maintaining security assurance cases at runtime were identified and validated through interviews with security experts or practitioners. These findings provided the general requirements for how the dynamic decision-support should be formed to in the end also increase the confidence in the security assurance at runtime. The results of the second iteration were then validated by security experts to find out how realistic they are.

4.2 Iteration 1

In the first iteration the main challenges towards assuring the security in a system at runtime were defined. This is strongly related to the first research question in the thesis. A literature review as well as interviews with security experts or practitioners in different industries were conducted in order to identify these challenges. The

interviews were used to validate the results that had been collected prior in the literature review.

4.2.1 Interview with Practitioners

After the initial literature review, semi-structured interviews with different practitioners were performed to gain further insights into the challenges of security assurance at runtime, especially more information directly connected to security assurance cases in practice, which were also relevant for the second research question in relation to the increase in confidence that more dynamic decision-support for security assurance cases would enable. The respondents were chosen based on their knowledge and experience with security assurance and the interviews were set up to last 45 to 60 minutes which took place either digitally or in person. It was difficult to find participants who have a sufficiently strong background in cybersecurity and security assurance cases in order to give meaningful answers. In the end, four respondents participated, which are detailed in Table 5.1.

The aim of the interviews was to identify the issues that engineers face when designing security assurance for their system and how this then plays out at runtime. Therefore, the questions were related to security assurance at design time, security assurance at runtime and how it affects decision-making as well as how runtime behaviour affects the existing security assurance. Finally, the last questions centered around how a more automated decision-support at runtime for unexpected behaviour could be integrated and how it would affect the confidence in the security assurance. The interview guide can be found in Appendix A.1.

The interviews were recorded with the respondent's permission. After the interviews, the recordings were transcribed and carefully read multiple times in order to fully understand the content. The analysis was conducted through a thematic analysis, which includes the transcription of the interviews, generating codes for the information that was collected as a form of labelling data fragments and in the end finding patterns and themes among these codes [42].

The transcriptions were coded as the first step in the qualitative analysis. The data was coded with open codes, which enabled a more explorative approach to find out about the challenges for maintaining security assurance cases at runtime. Open codes are codes that emerge during the coding process and are not set prior to the analysis such as with structured codes. An example of the coding process can be found in Figure A.1.

After each round of coding for an interview transcript, the codes were then reviewed to make sure that the same ones were labeled the same way and not repeated in different forms. Moreover, each code was assigned a value of how many times it had been referenced in the interviews and in the end also by how many interviewers. Before identifying themes, the codes were also organized into general categories based on the questions that were asked during the interview such as current practices,

challenges and solutions. This enabled a first overview of all the codes and how they connected across different interviews in order to also understand their relationship to each other.

The codes were then grouped together based on themes that could be identified in the data. A visualization of the process is also shown in Figure A.2. This took place through a coding workshop together with my supervisors where the codes and their themes could be discussed and evaluated. Both the coding and subsequent identification of themes took place using the online platform Miro. For each theme, the findings were further examined based on which interviewees had mentioned an aspect within the theme and the respondents' background. In the end, even though the majority of respondents are or had been mostly active in the automotive industry, some had insights from another industry or a different role in relation to security assurance which helped give further insights.

4.3 Iteration 2

In the second iteration, the focus was on how a security assurance case can be extended with game theory to enable more dynamic decision-making at runtime. In order to simulate how security assurance cases could be extended with more dynamic decision-support, an example of a security assurance case was created. The case is shown in Figure 2.1 and was used to base the artifact on that was developed in this iteration. The extension was evaluated by security experts in order to assess how the confidence of the security assurance case is impacted. Therefore, this iteration related to the second and third research questions.

4.3.1 Security Assurance Case Extension

The aim of the thesis was to understand how game theory can be integrated in security assurance cases to enable more dynamicness in the case and facilitate runtime decision-making during an attack. In order to show how a security assurance case would need to be adapted, an example assurance case model was extended with the necessary components to illustrate what the extension would need to include.

As discussed above, the example security assurance case in Figure 2.1 was used as the basis for the extension on which the design choices for the security assurance cases were made. The case is a simple representation of a fictional mobile banking app. At the top, the top-level claim regarding the system's security is made. A context node is connected to this in order to provide more information on the claim, such as the requirements for the system to be acceptably secure. Additionally, arguments are made based on the assets and potential threat paths in the system. At the end, the evidence for the claims is provided. The concept of the extension was created in Miro using the example security assurance case.

4.3.2 Validation with Practitioners

After the extension had been created, individual focus groups were conducted with each of the security experts from the initial interview in order to evaluate the extension's potential for increasing the confidence provided in a security assurance case as well as how effective it would be in a practical context.

The focus groups lasted around 60 minutes. During each session, the extension was shown and the idea as well as aim behind it was explained. Additionally, a small-scale simulation of the security assurance case extension in an attack scenario was shown to more dynamically illustrate the idea and give an example of how it might work in an attack setting. The simulation was built using Python. The attack scenario that was chosen to be modelled in the simulation was a probe attack performed by the attacker to which a security team can react to with either deploying a decoy system or not. The scenario is visualized in the simulation game tree shown in Figure 5.6.

After showing the artifact, the respondents were given a set of eight closed questions based on the Likert-scale, which were then followed up with four open-ended questions to enable more discussion around the artifact. The questions can be found in Appendix A.2. The focus of the questions was on the effectiveness of the extension, its usability as well as its impact on the confidence in both the extension and in the security assurance case as a whole as well as the system it represents. The use of questions in both a closed and open-ended format enabled comparability between respondents with the closed questions, however also more detailed information with the more open questions.

The results of the focus groups were afterwards analysed. For the responses of 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 was performed for the discussion-based questions in order to identify common themes among the answers and gain further insights into what aspects of the runtime adaptation of a security assurance case are important. Similar to the analysis in the first iteration, the data was coded with open codes and afterwards grouped into themes.

4.4 Validity and Reliability

Different measures were taken in order to maintain a sufficient research quality. These are detailed in the subsections below.

4.4.1 Internal Validity

In order to ensure the internal validity of the research, each interview with the security experts was conducted with an interview guide, so that the same set of questions were asked. Moreover, the use of closed questions in the evaluations also enabled

higher consistency in the data collection.

Furthermore, the sampling of respondents was based on people that are knowledgeable and experienced within security and security assurance cases. Therefore, this ensures the collection of data that brings useful insights and that is relevant to the research.

4.4.2 External Validity

The sample size for the primary data collection consisted of four security experts, which could be considered rather small and cause an issue in terms of the generalisability of the results. However, the goal was not for the respondents to be representative of their population and contribute to empirical generalisability, but rather to gather information in order to make theoretical generalizations such as for the challenges with maintaining security assurance cases at runtime. Consequently, by ensuring a high quality in the qualitative analysis, generalizations in terms of the theory can still be made.

Furthermore, the type of game and strategy used to model the runtime behaviour and interaction of an attacker and the system, which represents one aspect in the security assurance case extension, was considered carefully in order to reflect reality as accurately as possible and provide relevant insights. Nonetheless, the details of this model were not in focus for this thesis and therefore the actual game-theoretic part of the model is a relatively abstract and simplified version.

4.4.3 Construct Validity

In order to minimize threats to the construct validity in the data collection, the concepts and key words that often came up in the interview questions, such as security assurance, and how they are used in the thesis were defined and discussed with the respondents before the interview to ensure that the understanding of these concepts matched.

4.4.4 Reliability

Finally, the reliability of the method was ensured through the transparency of the reporting of the research method as well as the findings. For instance, a coding example is given to illustrate the analysis process and the interview guide that was used for all four interviews can be found in the Appendix. In addition to this, the thematic analysis for the interview in the first iteration was performed in a coding workshop with supervisors present which helped reduce potential bias that might have been introduced in the initial coding and therefore also helps ensure a higher internal validity of the research.

5

Research Findings

This section describes the results of the research. First the findings of the interviews with security experts will be described. Moreover, the decision-support model to include runtime adaptation in security assurance cases is presented as well as its potential effect on the confidence of the security assurance based on the results of the validation with the security experts.

5.1 Challenges of Maintaining Security Assurance Cases at Runtime

The first research question in this thesis focuses on understanding the challenges towards maintain security assurance at runtime. The challenges were identified through a thematic analysis of the interviews with practitioners. A visualization of this process can be found in Appendix A.3. The experts that were included in the study and their respective industries as well as interview mode and length are summarized in Table 5.1. In the end, four experts were interviewed who have extensive knowledge in the area of security assurance cases in practice. The experts have worked or are currently working in different industries as well as domains, which also helped identify challenges in multiple contexts and from different perspectives.

Table 5.1: Interviews with Practitioners

Interviewee	Industry	Current Role	Date	Interview Mode	Length
I1	Automotive	System Architect	7/3/2023	In Person	50 minutes
I2	Medical, Self-adaptive systems	Senior Researcher	8/3/2023	Online	50 minutes
I3	Automotive	Lead Engineer	13/3/2023	Online	50 minutes
I4	Automotive	Process Designer & Architect	14/3/2023	Online	50 minutes

Figure 5.1 shows the general themes that resulted from the qualitative analysis described in the previous section. In the end, seven themes or challenges were identified, three relating to design time and runtime respectively as well as one theme representing the transition between the two. The issues will be described in more detail in their corresponding section, also indicated in Figure 5.1.

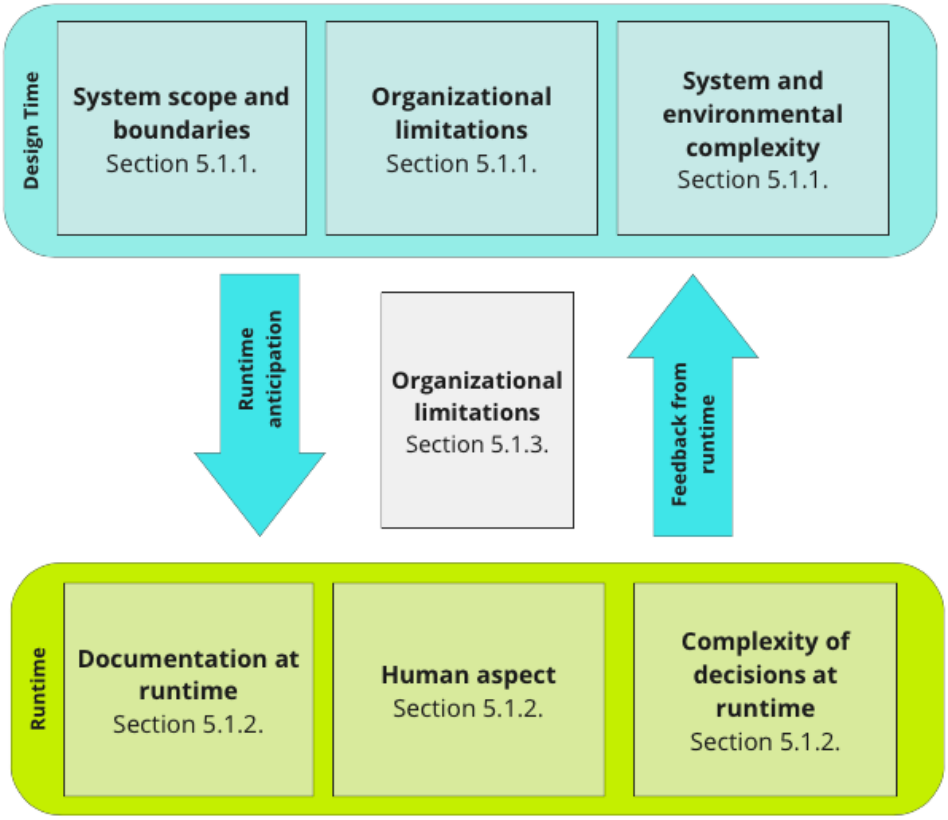

Figure 5.1: Main Challenges for Maintaining Security Assurance at Runtime

5.1.1 Challenges at Design Time

One of the challenges that was identified was the system scope and boundaries that are defined at design time. During the design of a system, engineers decide on what requirements the system and especially its security need to fulfill. A prominent issue mentioned by all interviewees is that it is difficult to cover all scenarios. Therefore, as it is difficult to cover everything, setting defined boundaries can on one hand facilitate the ability to account more extensively for possible uncertainties in the defined scope. However, considering that only mechanisms within the scope and the defined security boundaries are accounted for, this makes it difficult to maintain security assurance at runtime, since there are uncertainties arising from the context that the system is put in that cannot be accounted for. Moreover, I4 mentions that security goals and requirements are designed based on the identified vulnerabilities and therefore the system scope is further limited by the designers' abilities to anticipate vulnerabilities, which again relates to the issue mentioned above of covering all possible scenarios. Due to the uncertainty and complexity, both I1 and I2 mention that assumptions about the system in connection with the environment and the context need to be made in order to define what security features have to be implemented in the system and which one would be covered by for example the environment.

One way to manage these problems according to I2 and I4 is to design the system and functionality of the system so that there is less uncertainty to account for. For instance, I2 views the self-adaptivity of a system to be a form of acknowledging the uncertainty in order to manage it. These design implementations are also then what the security assurance will be based on. While strategic design choices are necessary in order to limit as many of the potential uncertainties in relation to the system as possible and therefore offer a solution for part of the issue, they still do not cover the context and unexpected behaviour from outside the system, which is not part of the security assurance documentation and leaves room for issues later at runtime. Therefore, defining the system's scope and boundaries present both a solution as well as still a challenge for maintaining security assurance at runtime.

Another challenge that was mentioned in the interviews are the organizational limitations that can interfere with the maintenance of security assurance cases at runtime. For instance, I3 mentions cultural challenges in companies such as too little value being placed on security assurance due to a lack of understanding of it. Oftentimes, testing is viewed as more effective than documentation of the system. This therefore points towards a lack of security awareness and subsequently security culture in companies. As a result, the focus on effective security assurance is decreased. Moreover, prioritizing new features over creating more effective security implementations, such as with security assurance cases, further leads to difficulties with the maintenance of the cases according to both I1 and I3. According to I4, there is also often a heavy reliance on a small number of security experts in a company to design the security assurance, where the underallocation of resources towards security assurance shows the lack of value placed on security assurance.

In addition to cultural and subsequent resource allocation-based issues, rigid and inflexible 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 both from new features or standards based on laws and regulations and problems arising at design time. There is often a slow incorporation of new standards according to I4. 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 security assurance cases.

Lastly, the system and environmental complexity is another challenge that makes it difficult to maintain security assurance cases at runtime. This was a challenge identified by all interviewees. Due to the complexity, there are multiple different uncertainties arising that are difficult to account for and react to at runtime. This results in the security assurance documentation being more static. According to I2, since systems operate in a complex environment and are often themselves very complex, it is not possible to consider all possible things that can go wrong.

I3 attributes the uncertainty of who the attacker is to the main source of all uncertainties in a system's security and thereby its assurance. They mention the

differentiation of complexity and resources of the attack to differ between a private individual compared to resource-intensive organizations. Therefore, the attack will be different depending on who the attacker is. Moreover, this affects the mitigation and the assurance needed at runtime as well as how security assurance is maintained, which are thus highly influenced by these uncertainties. Another big issue mentioned several times by all interviewees was the difficulty of covering all scenarios at design time. Being able to plan and account for all possible threats is difficult due to the dynamic threat landscape and again the uncertainty stemming from not knowing who the attacker is. All interviewees mentioned some form of threat analysis that takes place in order to get more insights into what might happen at runtime.

> *"The uncertainty comes from who's the attacker going to be. Is it going to be the bored kid in his basement just experimenting with his stuff, in which case that helps narrow down where that attack might come from. Or is it going to be like a nation state level actor or a major company that wants to take you down a peg and get at your secrets, in which case they're gonna come with a much more well hidden and manicured attack, specifically to get something out of you."* - I3

Finally, I2 brings up the observability issue that exists due to the dynamic nature of systems and security, where some components of the system and its context can only be observed at runtime and not properly anticipated. It is considered only from runtime observations that you can gain the complete understanding of the system. In fact, all interviewees mention the importance of verification of the assumptions made at design time and therefore the security assurance at runtime. However, both I1 and I2 mention the necessity of testing in the actual context in order to get relevant results. However, I3 says that it is often difficult to test all aspects in the system and therefore also the claims in the security assurance documentation.

I1 states that the main issue is the uncertainty of knowing how to act. According to them, security is dynamic, which is why it is difficult to predict and maintain everything. The uncertainty of anticipating what might happen in reality and therefore what the effective response is, was also an issue brought up by the other interviewees and therefore affects the security assurance from both the system and high level perspective. This also illustrates the necessity for more runtime support in security assurance to handle potentially unexpected behaviour.

5.1.2 Challenges at Runtime

At runtime when a system is faced with a threat, the severity and impact of the threat is first assessed. The teams tend to work with playbooks that give information on what action needs to be taken in a certain type of threat scenario. If the issue is more severe and immediate action needs to be taken, an incident response team will work on mitigating the issue. The challenges that arise at runtime are therefore based on the different runtime events and processes. One challenge for maintaining security assurance at runtime that is directly connected to events that

take place at runtime is the documentation in itself. Security assurance cases and security assurance documentation in general are typically very static and are not able to adapt to what happens at runtime such as unexpected behaviour. Therefore, while the complexity of the environment makes it difficult to plan at design time as mentioned in the previous section, I2 points out it would also not necessarily be possible to incorporate the dynamism even if it 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 often requires more people involved to take the right actions which ends up slowing the response down. Finally, new evidence is therefore also not dynamically accounted for, causing decisions to be less informed.

> *"However, 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 playbooks to be able to act fast. Therefore, playbooks should include solutions for quick decisions so that when the potential problem is not covered at runtime, actions can be taken as effectively as possible."* - I1

Another issue at runtime is the human aspect that goes into the analysis and the decision-making process when faced with an unexpected issue that was not accounted for during design time. I2 and I3 consider attacks to be a learning process where the exploitations of vulnerabilities will give information on how to make the system more secure for future attacks and drive new feature development. I1 mentions that there is always the possibility for human error both in the analysis of the attack and the decision-making. I2 considers system decision-making, such as in self-adaptive systems, a more reliable option, where the human error becomes much more unlikely. As mentioned earlier, they consider self-adaptive systems as a way to acknowledge and manage the uncertainty a system faces. However, the actions and scope in these systems are still defined by humans, which also relates back to some of the challenges regarding system scope and boundaries mentioned in Section 5.1.1. 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 according to both I2 and I4.

> *"Ideally you would learn something from each attack, that would give you some information about your system and you could make it more robust the next time around."* - I2

Furthermore, I3 mentions that 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 or the company. Therefore, many decisions tend to move across multiple levels in a company and between different teams. The complexity in the multi-level decision-making as well as the lack of agility in the communication processes when a certain hierarchy is followed slow down the decision processes and the overall ability to quickly respond

to attacks. Security assurance cases are intended to be the mean to simplify and facilitate the communication, however that is not always the case.

Moreover, as decisions move through multiple teams and levels in the company, domain knowledge of security tends to be lost as the decisions are not only influenced by the incident response teams with the direct knowledge but other engineers with other technical expertise and focus. This further impedes the incident response and increases the possibility of less ideal outcomes when decisions are made with insufficient information on the relevant contexts of the attack as well as if goals among the decision-makers are not aligned.

5.1.3 Challenges in the Maintenance Process

The maintenance of security assurance is another important part of its life cycle. The size of updates to the security assurance cases tends to vary based on the impact to the system. Moreover, the urgency of updates is usually in relation to how important the addition is to the security assurance cases and it providing instructions to security teams, according to I1 and I3. Additionally, I2 and I3 mention how the impact of the update is also dependent on how much of a "ripple effect" it has on the assurance case as a whole. Furthermore, I3 mentions that ideally the security assurance would be updated in connection with runtime analysis, which would also help speed up updates. However, according to I1, it is uncommon to update security assurance at runtime.

The aforementioned challenges show that the system at runtime and design time affect each other and the two states need to stay informed of each other. All respondents mention multiple times that there should be a strong interconnection between design time and runtime. A part of enabling this are continuous maintenance and updates of security assurance as well as effective organizational and workflow structures. However, the main issue that was identified to hinder the update and maintenance process, and therefore the transition of security assurance cases from runtime to design time and vice versa, are organizational limitations. This challenge had also been identified at design time as it is an obstacle towards establishing effective structures within companies, but in this context it refers to the organizational issues related to the dynamic feedback loop between the security assurance's states at design time and runtime. Therefore, challenges relating to the transition between design time and runtime also affect the maintenance of security assurance cases at runtime.

One of these challenges are uncertain maintenance processes. According to the security experts, there is generally uncertainty around the maintenance processes of security assurance cases. I4 says that companies are usually not well organized around maintenance processes and they should be integrated as part of the general process. In addition, according to I3, the lack of high-level structure often causes for instance security controls to be unnecessarily duplicated. Moreover, another effect of the uncertainty is the slow incorporation of new standards, which now also require

continuous maintenance.

> *"I don't think companies are well organized around this issue yet, but I think it's going to look quite different in different companies, especially about this part. Even on the part of doing a TARA (Threat Assessment and Remediation Analysis), people are not that well organized yet because it's a very new standard and there are so many things now that companies need to take care of around the processes and security etc."* - I4

Furthermore, the presence of designated teams for the maintenance of security assurance cases varies between different companies. According to I1, there are maintenance teams in charge of the changes, on the other hand according to I3 and I4 it is usually the development teams for each product that maintain the security assurance. Additionally, there is a lack of traceability and version control in regards to security assurance cases, which further makes it difficult to maintain them at runtime. According to I1, version control is not very common in the maintenance of security assurance. Therefore, this poses another issue to maintaining security assurance at runtime.

Another challenge in the maintenance process of security assurance cases is ineffective communication. I3 mentions that there are often issues around the communication of updates. Furthermore, I3 also mentions how the effectiveness of communication is connected to the organization overall and that organizational changes often cause issues in communication. Moreover, according to I1, changes to security assurance cases usually involve multiple forums, which can impact the speed and efficiency of the communication. However, overall this tends to vary between companies and their workflow structures and processes.

5.2 Potential Impact of Dynamic Decision-support

In relation to the second research question, the security experts were also asked in the interviews about how a dynamic decision-support model incorporated in security assurance cases would be able to increase their confidence in the security assurance as well as what requirements this model would need to fulfill in order for it to increase their confidence. Figure 5.2 shows the resulting themes as well as how they correspond to the challenges documented in the previous subsections and could potentially help support the decision-making at runtime.

The themes represent requirements that define how and whether a decision-support would help increase the confidence in the security assurance cases and help bridge the gap between design time and runtime by addressing some of the previously identified challenges.

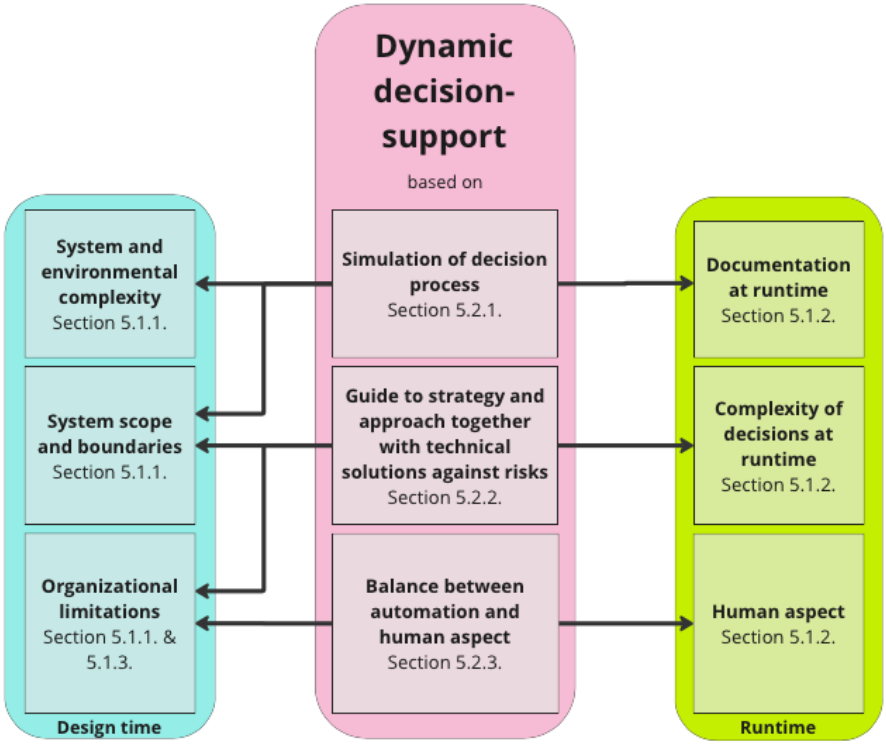

Figure 5.2: Dynamic Decision-support and Confidence in Security Assurance

5.2.1 Simulation of Decision Process

A dynamic decision-support would be able to aid runtime decision-making by simulating runtime behaviour and giving an assessment of the impact of a threat. By being able to assess the impact of the threat, it also indicates the severity of the issue, according to I1. Additionally, the respondent mentions that it would be useful if it could propose future actions to take in order to mitigate a threat.

Furthermore, I4 thinks that it would be useful if the decision-support could also visualize the decision process that typically takes place at runtime, i.e. how a security expert might think in an attack situation. This could further help increase the confidence for the security assurance.

5.2.2 Guide to Strategy and Approach together with Technical Solutions against Risks

In order to increase the confidence in security assurance cases at runtime, the interviews showed that it is important that a dynamic decision-support can help guide the strategy and solutions for runtime decision-making. According to I3, this would help facilitate the process and the discussions regarding what decisions and implementations to make, since it could help highlight the risks of different solutions. Additionally, the respondent mentions that it could help identify needs for specific knowledge, which also relates back to the issue of domain knowledge being lost when

decisions move through multiple levels. Moreover, according to I2, a more automated decision-support could help compile the knowledge about the operational conditions and environment to help anticipate runtime behaviour and guide decision-making also at design time in the design process.

5.2.3 Balance between Automation and Human Aspect

There is the potential human error both in the design of a system and at runtime during analysis and the decision-making. Therefore, a more automated decision-making would increase the robustness of security related decision-making and also reduce the human error. However, it is still important to make the process understandable for humans and preserve the transparency of the support mechanism so that engineers are able to interpret the process and maintain confidence in the system, according to I3.

5.2.4 Impact on Confidence

When asked about the potential increase in the confidence in security assurance cases with a dynamic decision-support, the participants' overall impression was that it could contribute to an increase, however this was also dependent on the different factors mentioned in the sections above. I4 also mentions that the impact on the confidence would be dependent on the target audience.

With a successful decision-support that is more dynamic, one of its potential advantages would be to reduce the impact on design time by having more dynamicism in security assurance cases to act at runtime, according to I2. Moreover, I3 considers the decision-support to potentially help communicate the value of security assurance cases and therefore increase the developers' perceived value of them as this was also linked to one of the organizational challenges.

5.3 Runtime Adaptation of Security Assurance Cases

In the second iteration of the research, the focus was on how security assurance cases can be extended with game theory. In a game-theoretic attack simulation, the interactions between attacker and the system are simulated, which is intended to be integrated in security assurance cases as a decision-support in order to make them more runtime adaptive.

The aim of the extension is, for one, to be 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 during an attack at runtime for humans, which can suggest mitigation strategies based on what actions the security team is able to take to defend the system against the actions of the attacker and might take in the future. For instance, in a certain attack, the security team might have

to decide between waiting for more information and relying on automated threat mitigation mechanisms or blocking the service that is being attacked. The optimal strategy is based on what the different payoffs for the system are when taking certain actions or strategies as well as the probabilities of what actions the other player might take. In the end, the strategy that has the highest expected payoff is the ideal strategy.

The proposed extension to the security assurance case is shown in Figure 5.3 and the full figure can be found in Appendix A.4. C3.1. is the overall claim representing the decision-support, which is connected to the argumentation strategy arguing 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 such as design features to keep the system secure, which 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.

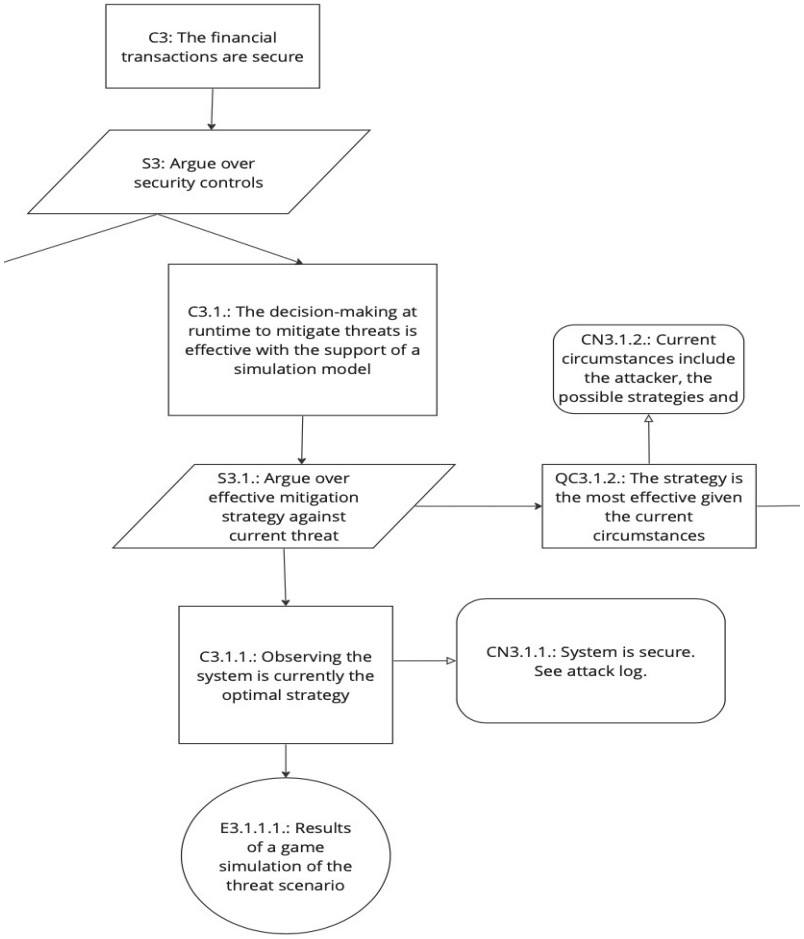

Figure 5.3: Security Assurance Case Extension: System is secure

In Figure 5.3, the system is assumed to be secure and therefore the current strategy would be to observe the situation, as shown in C3.1.1.. C3.1.1. finally results in the evidence I3.1.1.1., which is the result of the game-theoretic simulation model that calculates what strategy to take. Moreover, C3.1.1. also has a context node CN3.1.1. connected to it in order to provide the user with the context to the claim being made and explain why it is being made. In this context node, the user is referred to a hypothetical attack log that would be able to indicate that there are no attacker actions that a security team would need to take action against.

In order to define and argue for the quality of the decision-support and consequently the claims being made based on the game-theoretic model, the quality claim QC3.1.2. is connected to argumentation strategy S3.1.2. The claim structure is shown in Figure 5.4 and justifies why the claims made regarding the ideal strategy at runtime are acceptably justified 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 adequate estimation of their payoffs.

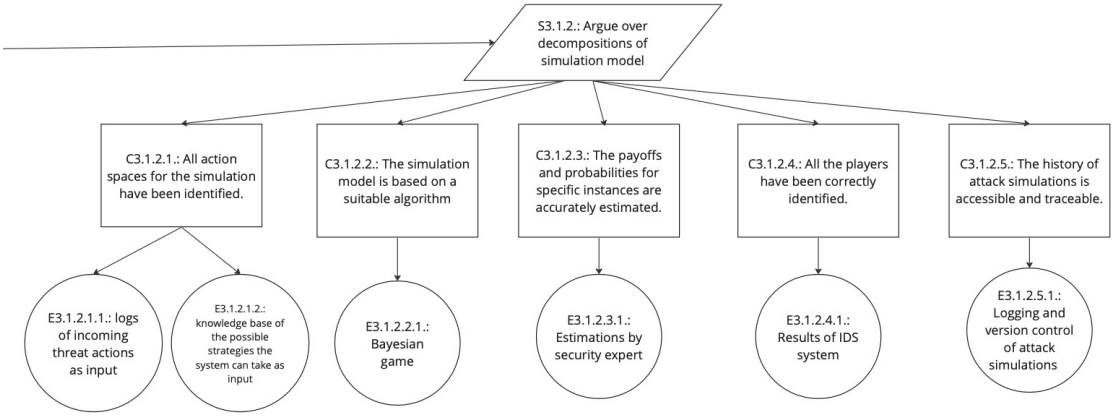

Figure 5.4: Security Assurance Case Extension: Quality Claim

Apart from the quality claim, the claims connected to S3.1. are dynamic claims that will change the output of what strategy is ideal based on what the game-theoretic simulation model proposes. Dynamic claims make it easier for the user to follow what action needs to be taken in order to mitigate a threat. They represent a current version of the system's security and what the case claims needs to happen in order to restore or maintain security. Additionally, given that there are usually multiple mitigation strategies available to a decision-maker at 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 5.5 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.

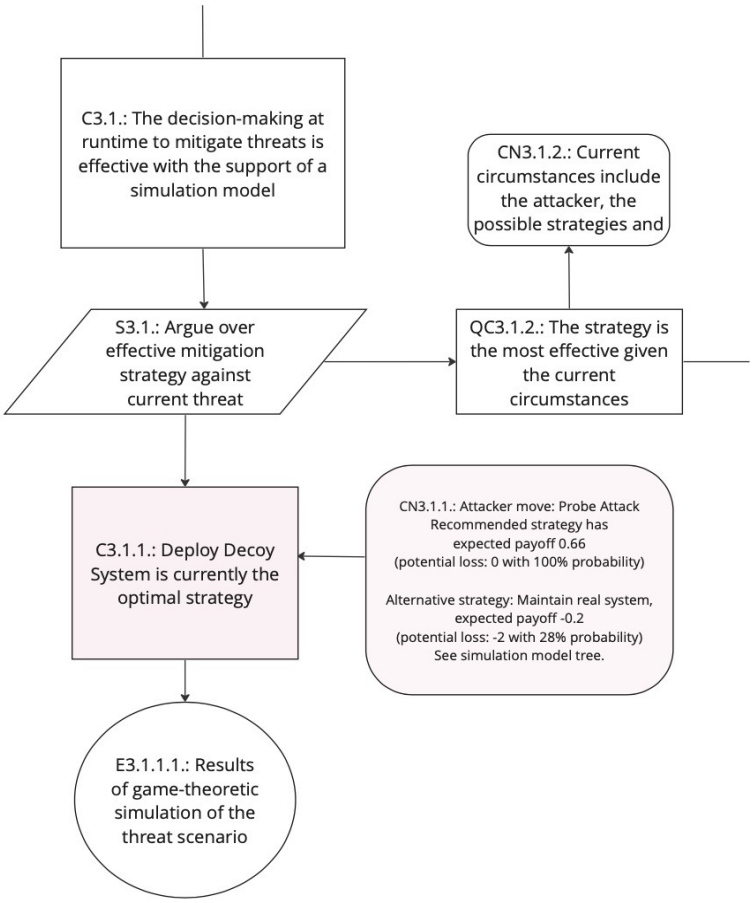

Figure 5.5: Security Assurance Case Extension: Threat to be mitigated

In addition to the claim, the context node will also change to represent the new circumstances of what strategy the claim proposes. The expected payoff and potential loss for the strategy in the claim are included as well as the alternative or next-best strategy with its expected payoff and potential loss in order 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 the context node relate to each other and thus enables them to assert their own judgement 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 also look at a visual representation of the game simulation and would be able to see an overview of all the different strategies and their payoffs. Therefore, this information would then be able to support the security team in making decisions at runtime and help reduce the potential human error from making their own assumptions.

Based on the interviews, many security teams tend to be based on specific products or version of products. Therefore, in the security assurance case the decision-support 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 security assurance case example, the decision-support claim C3.1. is added in relation to the financial transactions asset C3. In practice the asset might be even further decomposed into sub components of the asset, which would make the possible attacks and mitigation strategies more detailed and limited in number. The exact placement of the decision-support's top claim C3.1. would therefore depend on the exact system, however it is important to maintain the claim at asset level as one of the security controls protecting the corresponding asset in order to maintain the claim's accessibility and subsequent usefulness as a decision-support at runtime.

Moreover, the decision-support's effectiveness is very dependent on the game-theoretic simulation model and how effective it is in suggesting the ideal strategy. The model's formation and algorithm was not the primary focus in this thesis and would in reality need to be improved. Instead, emphasis was placed on identifying the requirements that this type of model would need to fulfill in order to support the decision-making. The necessary aspects in the game model will be further explain in Section 5.3.1.

5.3.1 Game-theoretic Simulation Model

The example game that is used in the simulation model to demonstrate the idea for the security assurance case extension is a Bayesian game with mixed strategies. The example attack that is 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 is shown in Figure 5.6, which is also the simulation model referred to in the context node CN3.1.1. in Figure 5.5. Based on the system's response there are different possible payoffs depending on the type of attacker as well as the possible mixed strategies of the different attacker types after the system has made its move.

The environment of a system can be relatively complex and is often subjected to multiple types of uncertainties, particularly in regards to security, that are difficult to account for in security assurance cases. Security in itself is dynamic due to a continuously changing threat landscape, where the uncertainty comes from both the technical complexities as well as the attackers themselves who have their own motivations. Moreover, the system that is being protected is also complex and dynamic with new updates and changes being made, which can open it up to the possibility of a new security threat. In addition, the human error is another potential threat both during the design of the system's security as well as during runtime analysis and decision-making. Therefore, it is important that the game-theoretic model that is created as part of the decision-support is able to account for some of these uncertainties and complexities, which is typically achieved with probabilities in order to

be able to incorporate the possibilities of different scenarios.

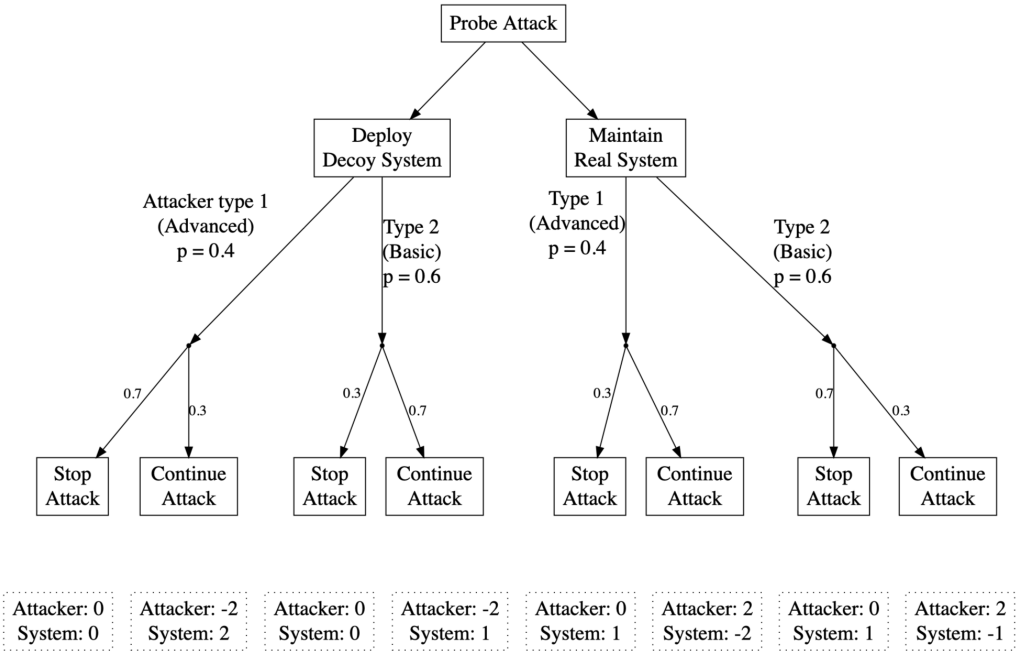

Figure 5.6: Game-Theoretic Simulation Model

Due to the uncertainty of what the actual attack might be as well as who the attacker actually is, there is a high degree of incomplete information that both the players have. Neither of them know what the other party is aware of, what exact strategies or actions they can take and one of them might also have more information compared to the other causing information asymmetry. 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 represented by a move from another player referred to as 'Nature' or 'Chance', which would be the events taking place outside the players' control to which they can then react to with their available strategies. In Figure 5.6, the game considers two types of attackers and subsequently attacks 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. Furthermore, 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 simulation or attack-defense models and also game-theoretic models in general. These implications will be further

discussed in the next chapter.

5.4 Impact on Confidence in Security Assurance Case with Runtime Adaptation

In order to answer the second research question of how game theory could potentially impact the confidence in security assurance cases, the security assurance case extension was evaluated by the four security experts that had also been interviewed in the first iteration of the thesis. The dates for the evaluation of the security assurance case extension as well as the format and length are summarized in Table 5.2.

Table 5.2: Focus groups with Practitioners

Interviewee	Industry	Current Role	Date	Focus Group Mode	Length
I1	Automotive	System Architect	28/4/2023	Online	65 minutes
I2	Medical, Self-adaptive systems	Senior Researcher	5/5/2023	Online	60 minutes
I3	Automotive	Lead Engineer	4/5/2023	Online	60 minutes
I4	Automotive	Process Designer & Architect	4/5/2023 & 12/5/2023	In Person & Online	90 minutes

The results of the focus groups consist of the answers collected in combination with the closed questions as well as the findings from the analysis of the discussion-based questions.

5.4.1 General Impression of the Extension

The respondents were asked both closed and more open-ended questions in order to get a general overview of what they all thought about the artifact, which could then be compared among them through the collected quantitative results as well as identify more specific advantages and issues of the extension for future improvements. All the results of the closed questions along with the central tendency of the data, consisting of the mean, median and mode, as well as the standard deviation are detailed in Appendix A.5. Moreover, the answers are also visualized in the graph in Figure 5.7.

The closed questions aimed to measure three aspects of the extension; its usability, effectiveness as well as the confidence it inspires in relation to a security assurance case, 1 being the lowest value and 5 being the highest. The specific results for the different areas are detailed in the subsections below. Combining the measurements for the different areas can give an indicator into how the experts valued the decision-support extension overall. The mean of all questions and answers lies at 3.438. Both the mode and the median for all questions and answers are at 4. Therefore, the answers of the respondents generally indicated that the extension is mostly effective.

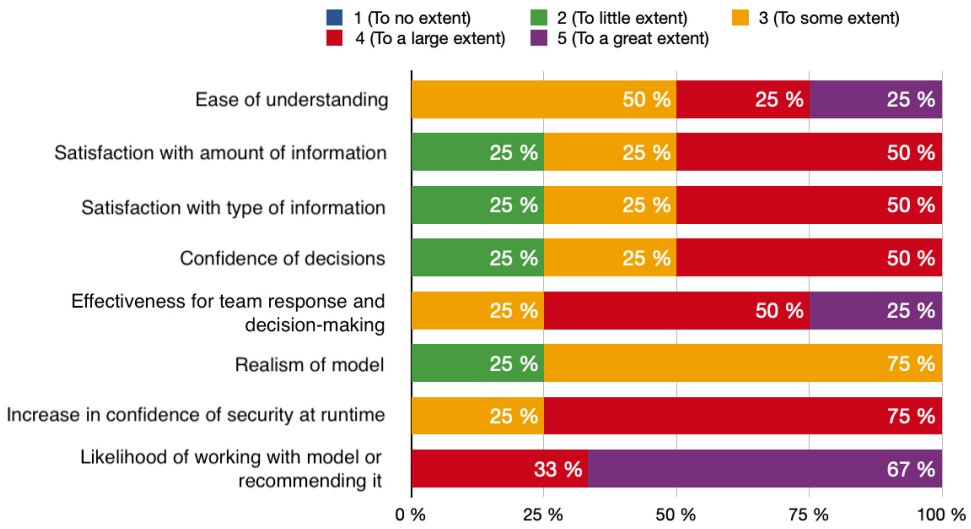

Figure 5.7: Results of Closed Questions from Artifact Evaluation, see Appendix A.2

5.4.2 Usability

Questions 1, 2 and 3 (Appendix A.2) were focused on measuring the usability aspects of the extension. The extension's ease of understanding was rated at an average of 3.25. Therefore, the extension was somewhat easy to understand, however most respondents stated that explanations and clarification were still needed. The context of the extension, i.e. combining game theory with security assurance cases, was initially unfamiliar for some respondents and therefore lead to some confusion. Furthermore, the functionality of the tool in relation to both game theory and the security assurance case, such as the quality claim or the use of a dynamic claim was also difficult to follow. According to I2, many of the concepts that were mentioned needed to be explained and were not very intuitive. I1 and I4 stated that their understanding of the extension increased over time with explanations. Moreover, I3 did not think they would be able to understand the extension without someone explaining it, however the model was easy to understand once they understood it.

Another usability measure was the average satisfaction of the type of information that was included in the extension, which was rated at 3.25 with the most frequent answer being 4. Therefore, the type of information was considered somewhat satisfactory. I3 based their reasoning on the fact that the user is required to have a certain baseline of knowledge, for instance in relation to game theory, in order to interpret the information that is given, which they assume most people would not have in this situation. Therefore, while they consider the type of information to be somewhat satisfactory, the decision-support requires a level of understanding of game theory that cannot necessarily be expected in this field. Moreover, I2 did not find the dynamic claim to be useful in this context as the connection to the available mitigation strategies was missing. According to them, the strategies would need to be part of the extension and in addition to what is shown in the simulation tree,

the strategies would also need to be stated explicitly in the security assurance case.

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. This measure had the highest mean of 3.75 among the usability related questions, but also the lowest mode of 3. Overall, respondents mentioned that they needed many explanations of the extension and also in relation to the game simulation tree and how that can be used. I1 mentions that they would have missed some things without the explanations and I4 says that without more clarification there is the risk of users having to make guesses. Moreover, I2 considered their answer to be dependent on further developments of the tool, specifically how available mitigation strategies are included in the case as mentioned in relation to the measure for question 3. Finally, according to I3, once the extension is explained they considered the type of information to be very satisfactory, where for example 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.

5.4.3 Effectiveness

Another aspect of the decision-support was its potential effectiveness in relation to confidence in the decision-making at runtime (question 4), how effective it can be in the team's overall decision-making and response (question 5) as well as how realistic the extension is and if it could actually work as intended in reality (question 6). The means for the answers varied between 2.75 and 4.

The extension's effectiveness for security processes such as the overall decision-making and response was rated at a mean of 4 and therefore shows that the extension is considered to be useful for supporting the activities of a system's security response. However, according to I2, while they see the usefulness of evaluating between different mitigation strategies, they are not sure whether security assurance cases are the best format due to their complexity and size that make them difficult to use especially at runtime. Moreover, I3 mentions that the incident response might arrive at a similar conclusion as the decision-support and therefore the extension might not significantly impact the effectiveness of their activities. However, the respondent bases their answer on how effective it can be for other users not directly involved with the incident response, but with the asset itself and that might also not necessarily be a security professional. With the decision-support, I3 thinks the extension would facilitate their participation in for instance discussions regarding mitigation of threats.

Furthermore, in relation to the ability to confidently make decisions at runtime with the extension, which not only reflects its potential effectiveness but also the perceived trust in the system's recommended strategy, was rated at a mean of 3.25. The dispersion of the answers was not high, however one of the higher values and equal to one of the usability related questions. Therefore, it is possible that it might depend on contextual factors such as how exactly the extension would be used. I4

stated that the structured format of the extension with the probabilities help 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 is meant to be supported with this thesis and also concepts such as payoffs and probabilities might be difficult to gauge for humans. Moreover, according to I1, an approach such as this extension would need to be used and tested in order to evaluate how reliable it is and therefore the confidence one would have in its support. On the other hand, according to I3, similar types of decision-support are already used only under different circumstances and therefore if the extension that is proposed had an effective game-theoretic model that is able to deliver recommended strategies in actual attack scenarios, it would be an effective guide.

Lastly, the question concerning how realistic the respondents considered the extension was rated the lowest of all questions with a mean of 2.75. In addition, the standard deviation was also one of the lowest, which also shows a form of consensus among the respondents in their answers. The respondents found it difficult to estimate how realistic the decision-support would be for different reasons. I2 considered the extension to be too simplistic and also raised concerns in regards to the increasing complexity of attacks in reality and how this would be accounted for as well as if that was even possible while keeping its purpose of supporting human decision-making. Moreover, I4 stated that it would depend on the attack and the attacker whether this extension would be realistic and therefore effective. Generally, the worst attacks are by attackers that are well-prepared and therefore happen very fast without time to react or a human to deploy a defending move. Therefore, there is no actual interaction between system and attacker in the sense of a game. On the other hand, I4 thought that the decision-support might be more realistic and effective with more advanced attacks that have deeper attack paths. I3 also mentioned that the type of attacker would influence the answer to this question, where very resourceful and advanced attackers that are very difficult to detect require a different form of thinking. Moreover, I3 mentioned that the realisticness and thereby effectiveness of the extension would depend on the context of both the attack scenario and what type of guidance or support it would provide. In some cases, it would be more useful if the extension provides decision-support in relation to actions to take to maintain security in a system at runtime when uncertainty increases rather than direct responses to an attack. Finally, according to I1, additions such as clearer mappings to the system impact are needed in order for the extension to be effective and a realistic support. However, when faced with an active attack that the security team is able to react to with pre-defined alternatives, I1 stated that this extension would be realistic and effective since the decision-support can evaluate the different options at runtime.

5.4.4 Impact on Confidence

The confidence was measured through the last two closed questions. In regards to how much the confidence of the security assurance case would increase, most answers rated the extension at a 4 with the mean being 3.75 with relatively low

dispersion in the answers. Therefore, the extension can help increase the confidence of a security assurance case, however it will depend on the circumstances. I4 finds that the extension can help consider the runtime aspects of a system and therefore help increase the confidence in the case. Furthermore, according to I1, the extension can provide evidence for the continuous monitoring activities in the system at runtime, which would add more validity to the claims in regards to the security of the system. Moreover, I3 mentions that the extension can help provide more automation to defining what processes to follow and that can help ensure continuous compliance. However, generally, I3 finds more purpose of the decision-support on the procedural level rather than the operational one. Lastly, according to I2, the increase in confidence in the assurance case would depend on what exactly a system can do on its own without human interference in terms of security.

The last question asked the respondents to rate the likelihood of working or recommending the tool where most respondents rated it at 'probably' or 'definitely'. I2 could not answer the question in the extension's current state, since in relation to self-adaptive systems the value would depend on the extension's purpose in this context and the role of the human versus the system. Therefore, the potential confidence of the extension can be considered relatively high, however there is more specification needed in relation to how it would work in more specific contexts, such as self-adaptive systems. However, other respondents considered the concept to be potentially useful after further development and testing.

5.4.5 Advantages of the Extension

In addition to the results from the closed question, the discussions with the security experts revealed some of the potential advantages of the extension. The respondents mention different aspects, where they think the decision-support and extension could help improve different security-related activities. For instance, I1 and I2 mention 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 mention that it could be useful in penetration testing, which could serve as evidence in the security assurance cases and thereby increase its confidence.

Moreover, I1 and I4 mention how the extension could be useful in training people and improving security capabilities of, for instance, the incident response. I3 and I4 also consider the extension to be helpful 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. Many companies do not have a fully operational or sufficient incident response and therefore a decision-support would be useful to also help guide the security processes on a more systemic level, according to I3. In addition to this, I3 and I4 mention that the extension can be understood and used by a broader user group, which might also reduce the reliance on the incident response at runtime.

Another aspect where some of the respondents considered the tool to be potentially useful is in the traceability and logging of attacks, which could further help in the learning process of the system's security in different attack scenarios. I1 considers the potential logging ability of the tool to be useful in 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 security assurance case. Furthermore, I3 mentions that the extension could possibly help identify some of the redundancies in for example security controls.

5.4.6 Limitations and Possible Improvements of the Extension

The results from the discussions with the security experts also highlighted some of the limitation of the extension, which are discussed in the subsections below and further explain some of the values discussed above.

5.4.6.1 Security Assurance Case Aspects

As explained in Section 5.3, 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 assurance case. The claim then decomposes into new claims related to the mitigation strategies that should be taken to resolve a threat. In addition to this, a quality claim was connected to the strategy. However, this quality claim and the structure of claims that follow it were confusing to some security experts as it was unclear how this claim related to the mitigation strategies, since there is a connection between them but only in terms of the logic behind what claims are being suggested. Furthermore, I2 also explains that the TARA is another form of evidence that should go into the simulation tree in the security assurance case, since it provides information that can be used as a starting point for the system's defense.

Another part in the security assurance case that was unclear was the phrasing of the idle strategy "Observing the system". I2 interpreted this as meaning the entire system, even though the decision-support only relates to a specific asset. Moreover, the context to the observing claim was also confusing as it is unclear whether the system might be observing itself or whether a human is doing it. In the context of the security assurance case, this raised concerns for how secure the current (idle) strategy actually would be, since it would depend on what can actually be observed, such as what the intrusion detection system can detect or overall what sensory input the system has access to.

Moreover, as mentioned in the analysis of the answers for the closed questions, I2 was missing more clarification for what mitigation strategies are available to a security team in an attack scenario explicitly in the security assurance case. Therefore, aside from the game simulation tree, the mitigation strategies would also need to be included in the assurance case to give a better overview of the choices that a

decision-maker would have available. Additionally, the use of a dynamic claim was difficult to follow as there is no information in the assurance case of what possible mitigation strategies are available at that state. Since the alternatives were not presented in the case, I2 explains that it is difficult to understand how an action is integrated with the system and more specific mapping between the available mitigation strategies to the specific aspect in the system is needed. I3 found it also unclear what baseline for strategy selection is used, in relation to the context such as what standard is used to guide them. Therefore, a more explicit connection between the available strategies and what is shown in the simulation would be needed.

Furthermore, in relation to the decision-making that the extension enables, I1 also explains that it would be useful to have more of the available options shown, aside from the alternative strategy in the context node, since it is important to be able to compare between different options. Moreover, according to I1, the payoff by itself in the assurance case is difficult to understand and whether that is a high or low impact, since it would need to be more clear if it reflects an attack on a very sensitive part of the system that might be more severe. I2 also explains that impact is usually expressed as categories. However, I2 mentions that the potential loss is a useful measure for decision-making. On the other hand, the use of layered probabilities might be difficult to follow, according to the respondent.

Finally, some respondents stated that the phrasing in the argumentation strategy S3.1. was confusing as it was not clear if the claims were targeting a specific threat or if the threat landscape overall was considered. I1 suggested that a context node should be added that specifies that all threats are considered in the strategy. Moreover, I2 mentions that it might also be necessary to be more explicit in who is mitigating the attack, since especially in self-adaptive system there is a clear difference between the role of the human and the system at runtime when faced with a threat.

5.4.6.2 Clarification of the Extension and its Context

All respondents mentioned frequently the need for explanations and clarifications of the extension and the game-theoretic concepts that were included. Therefore, on the general level and in order to understand the extension as a whole, I1 mentions that high-level definitions of the concepts and use cases would be important to include. I1 also explains, since the simulation tree consists of a large amount of dynamic data, it is important to explain how they change in order to facilitate its usability. Moreover, I3 mentions including a guide to explain the extension. However, I1 and I3 stated that the game simulation made sense and I4 also mentions that decision trees are a common visualization tool for decision-making processes and therefore would be understandable for most people.

Furthermore, on the specific aspects of the game-theoretic model and using it, I1 expressed, for instance, 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 as well as understand when these values end up changing

due to new incoming information. In addition to this, I1 mentions the importance of also 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 in order to further assess the severity of the issue. In the extension, it is therefore difficult to tell which part or parts of the asset, that the extension connects to, are affected, how the asset is affected, i.e. how "deep" the attacker is in the system and therefore the extent of damage an attacker can do, as well as how this is reflected in the payoff in order to be able to make decisions with confidence. I2 also mentions 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 both in terms of the value of the asset itself, its level of degradation and also to what extent it is compromised in order to better assess the severity of the issue. Therefore, having a singular payoff value that is an aggregation of these aspects is not sufficient.

5.4.6.3 Perspective of Attacker

In the evaluation, the experts 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. Moreover, according to I1, one aspect to threat modelling 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 team would benefit from certain actions. Therefore, the attacker payoff might play more of a role in the decision-making, especially in quicker decisions, in how to mitigate an attack and would therefore need to be included in for instance the context node as well.

Moreover, I4 mentions that the attacks at runtime do not always play out as games, where the system would have a set of action to counteract the action of the attacker. Usually, the first 'move' by the attacker is the attack itself, which then has to be resolved by the security team in the aftermath of the attack. Therefore, how security experts reason around attacks might not typically be in the form of a game-theoretic interaction and could be a limitation in terms of how realistic and effective the decision-support would be.

5.4.6.4 Scalability

The scalability of the extension was a concern raised by some of the respondents, where they were not sure how exactly the decision-support would work in an actual attack scenario and in the end how usable it would still be when the number of choices increase and more complex simulation trees might arise. I1 mentions that if they had to make an immediate decision, a decision tree that is too large would not be helpful. They also mention that more additions to the current version of the extension would require more views and separate files to be referred to, rather than adding more information to, for instance, the context node. Additionally, I1 and I4

mention the importance of illustrating the extension with more realistic examples as well as performing, for example, penetration testing to see how it would actually behave and work in a more realistic setting.

Moreover, the security experts considered the overall realisticness of the model to be dependent on different contextual factors, such as what it would be used for and by whom. One aim of the extension is to help guide decision-making and enable a faster response when faced with an attack. However, there are also concerns with how usable the security assurance case would actually be in these types of situations. For instance, I2 expressed doubts regarding the use of a security assurance case for runtime decision-making, especially in cases where there is little time, since the assurance cases can sometimes be too complex to navigate and therefore impractical to enable fast incident response. On the other hand, I2 mentions that it would depend on how incident response teams operate and how they train for attack situations. Moreover, respondents I1 and I4 consider security assurance cases to be potentially useful also in this regard. Furthermore, I3 stated that the extension would be most useful for security decisions a more systemic level. Therefore, the context of how the decision-support is implemented and used as well as its actual user affect how effective it would be and in turn how it can impact the confidence of the security assurance case.

6
Discussion

In the following subsections, the empirical findings for each research question are discussed.

6.1 RQ 1: Challenges of Maintaining Security Assurance Cases at Runtime

Multiple issues were identified to being able to maintain security assurance cases at runtime and being able to use the cases as effective support for decision-making when faced with a threat. These issues could be divided into challenges taking place at design time, runtime or in the transition between these two states. One of the issues relating to the system and decision-making at runtime is, for instance, that the security assurance cases are very static and would need to incorporate some form of dynamicism in order to adjust claims and evidence to aid the decision-making. Other challenges that directly relate to the runtime specific issues are the potential for human error during decisions as well as the complexity of the decisions that have to be made. These factors can be reduced and potentially mitigated by extending security assurance cases with game theory and including a dynamic simulation of the threat scenario at runtime. Moreover, the challenges that take place at design time, and also impact the runtime effectiveness of security assurance cases, are the system and environmental complexities as well as potential restrictions from the system scope. Uncertainty was considered by for instance Weyns et al. [17] and De Lemos et al. [18] as the main challenge for security assurance cases 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 these taxonomies, only in this thesis 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 security assurance cases as this issue does not necessarily stem from uncertainty. Nonetheless, the main issue overall remains the high degree of uncertainty in security and how it is managed in organisations, which then causes these individual challenges to arise.

The uncertainty that arises due to the complexities in the system and its environment can be managed with game theory as it is able to consider the uncertainties with probabilities of different possible actions of play, such as through stochastic games or the use of mixed strategies. This can then support runtime decision-

making and reduce other challenges such as possible human errors in the analysis or decision-making process. By also managing the complexities related to the system, the decision-making chain within the company as a whole could be simplified, if the model incorporates the different factors to consider and weighs them against each other when suggesting the optimal strategy, for instance through the payoffs in a game-theoretic model.

One challenge that can be difficult to manage with only a security assurance case extension might be the ones related to the organizational processes and culture. While a more effective security assurance case that is able to consider actions at runtime can end up receiving more positive reception from security teams in the company, it is still a broader challenge that would involve active participation of stakeholders such as management in companies in order for a runtime adapted security assurance case to be able to take effect and support decision-making at runtime. The maintenance of security assurance cases along with the system they represent is essential, especially in more security or safety-critical contexts, in order to be able to provide assurances of certain system requirements to be met [7]. Therefore, aside from tools that facilitate the maintenance and enable more continuous compliance, it is also central to have effective organizational processes that further enable the maintenance.

6.2 RQ 2: Impact of Game Theory based Decision-making on the Confidence in Security Assurance Cases

The initial interviews as well as the evaluation of the artifact provided useful insights into how the inclusion of a dynamic decision-support can impact the confidence in security assurance cases. The security assurance case extension in this research is on a rather conceptual level and would need to be actually implemented in order to put the ideas and assumptions in a more practical context, which can then be tested and further evaluated. However, based on the results of the evaluation, the concept in itself needs to be developed in order to become more effective and useful as well as easier to understand for practitioners. As I1 mentioned, the system impact needs to be integrated further in order to have more context on both payoff values and the attack so that humans' evaluations and subsequent decision-making is better supported. Moreover, the confidence in the security assurance case could potentially increase with the decision-support. However, this is dependent on different factors, such as how easily the case can actually be understood in reality.

Overall, respondents said they would find it difficult to understand the extension in the security assurance case without explanations or context. The idea of making security assurance cases more runtime adaptive by including game theory in order to then also support runtime decision-making is a new and complex concept. Therefore, an executive summary or this thesis would need to come in connection with the extension. However, the case shown was simplified and also focusing on the

extension of the security assurance case, not giving details on existing claims that were not related to the extension, which might have made it difficult to understand how the claims related to the decision-support fit with the remaining assurance case. Moreover, the sessions were only around one hour and given that the respondents understood the aim and idea of the extension within that time, albeit with some context and in the future with more written indication, it would still seem possible to understand the model as long as there are explanations in the form of an executive summary, guide or use cases. Therefore, it can be considered to add more explanations to, for instance, the simulation tree, however that would need to be weighed against how much information it should contain overall, since this is a simple version of an attack and in reality with more choices, too much more information can become too complex for a human to understand.

Aside from the general context of the decision-support and its features, the importance of the system impact and mapping to it were also brought up. The payoffs were intended to represent an aggregate value of the impact of an attack on the asset that it is protecting. However, based on the results of the focus groups, a differentiation needs to be made regarding the asset's value to the system and to what extent the value is affected by the attack as well as potentially the strategy that is chosen. Moreover, how the asset is affected as well as the system overall, i.e. their degradation, would also need to be included and differentiated in the payoffs.

Another aspect in regards to the decision-making was the mapping of the ideal strategy to the available mitigation strategies. As discussed in the previous chapter, the extension suggests the optimal mitigation strategy through a dynamic claim, which can maintain the assurance case in tact with the current state of the system. However, some respondents would prefer a clear mapping of the available alternatives in order to understand what is possible and be able to weigh all the options against each other. Therefore, the extension might require too much trust of a human to take the recommended strategy of the decision-support as it is. The motivation behind having a dynamic claim is to clearly reflect the system's state at runtime, where in a specific attack scenario there would only be one set of actions to mitigate the attack. Assurance cases can be rather complex and difficult to use at runtime, especially when fast decision-making is required. The complexity of assurance cases, especially when all information surrounding the security of a system needs to be accounted for, was pointed out by multiple respondents. Therefore, having one claim that can dynamically change to show what to do might help facilitate the use of the assurance cases in these situations. The overall need to be able to follow the decision-making through more context either through seeing all available strategies or being given more details regarding the payoffs, relates back to the theme of balancing the automated and human aspect of a decision-support that was identified in the interviews in the first iteration. However, it appears a trade-off might have to be made between the amount of information a security expert would require to make decisions in relation to attack mitigation while still maintaining low complexity and high accessibility to the necessary information.

Furthermore, the evaluations showed that the decision-support might not be particularly effective in connection with self-adaptive systems, since humans in this context usually take on a different role which also translates to the security assurance case. There is less of a need to manage a system at runtime, since self-adaptive systems are a mean to manage the uncertainty in their environment [18]. Moreover, due to the typically high pressure on humans in attack scenarios at runtime, the complexity of decision trees as well as security assurance cases might be too large and therefore not ideal to use during runtime decision-making. Therefore, in some instances the decision-support might instead be more suitable as a form of validation of the design in terms of its effectiveness at runtime. This was also brought up in focus groups as an advantage of the decision-support proposed in this thesis, as they mentioned the extension's potential usefulness in the validation of security measures and the security assurance case as a whole. According to Rushby [12], tools for automated decision-support can help reduce potential confirmation bias that can arise when designing features since it could help point out issues that had not been identified earlier. Therefore, the extension can be effective in the evaluation of security assurance cases. The traceability that the extension could help provide from attack situations was also considered useful by the security experts, which can then be used to connect evidence with security requirements and help increase the validity of the assurance case [7].

Finally, many of the advantages or potential use cases of the extension mentioned by the security experts were generally not in relation to the incident response, but rather the process aspect to incident response and using the decision-support for other, more design time focused activities, such as a guide to compliance as well as threat monitoring and lastly training. It is possible that the extension is not particularly usable at runtime due to for instance the format of security assurance cases. However, the security experts in this research are also more active in security domains that are not related to incident response and some found it difficult to speak on how some processes in this area take place. Therefore, it is possible that more insights need to be gathered into how the extension could work in relation to the incident response against active attacks.

6.3 RQ 3: Security Assurance Case Extension with Game Theory

The security assurance case extension was presented in the previous chapter, where the decision-support and its related claims are proposed to be added as a security control in connection with the asset in the case. However, there is also the trade-off of giving more specific insights into what level the decision-support is operating at in an asset, while still maintaining accessibility and thereby usability for the decision-maker. Another alternative that was considered during the development of the extension was connecting the decision-support at the attack level in the security assurance case 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. Therefore, it would be difficult and in some cases it might not even be possible to correctly connect a mitigation strategy proposed by the decision-support to a specific attack. As a result, it is more suitable to extend the security assurance case on the asset level to make the decision-support more effective at covering multiple scenarios. Moreover, the assurance case might become too complex, since the threat landscape is very dynamic with many different types of attacks that will also further vary depending on what type of attacker is encountered.

Another alternative that was considered besides the asset level was to connect the decision-support to the claims and arguments that relate to the processes that take place at runtime to protect the system, such as the incident response. However, these processes can vary significantly between companies and a more abstract version of the extension was considered more realistic in practice. Moreover, the strategies that would be proposed by the decision-support become more understandable in connection with a specific asset as that is what the attacker is targeting during a threat. The decision-support in itself is a security control connected to an asset and gives strategies to mitigate threats against the asset. Therefore, it is in terms of the process perspective only a tool and does not give a step-by-step for their actions or decisions in relation to the process.

Overall, the implementation of the extension is similar to the security assurance case adaptation proposed by Jahan et al. [6] for self-adaptive systems, since the extension operates similarly to a MAPE-K loop in the sense that the system and its environment are continuously monitored in order to provide the information that is then analysed by the game-theoretic model to give a suggested mitigation strategy. The continuous monitoring of real-time data is necessary for the model be able to make accurate decisions at runtime. However, the execution step would be decided on by a security team, instead of only the system. Therefore, the extension in this thesis differs from the security assurance case claims and context nodes suggested by Jahan et al. [6], in that the claims in relation to the decision-support are expected to be used by humans and therefore include more information to enable them to make their own risk assessments.

Apart from the security assurance case, the runtime adaptation is only able to inspire confidence if the game simulation in itself is also effective. In this thesis, a relatively simple model was used to simulate the extension which considers some uncertainties of an attack scenario such as the attacker type, however in practice a more complex model would be needed that is able to consider the uncertainties and impact of choices on a more widespread level. Therefore, another viable option would be stochastic games such as Markovian games or Markov decision processes. However, since there is a high degree of complexity and uncertainty involved with attacks, some form of probabilities should be included as they are a mean to manage the uncertainty at runtime.

6.4 Limitations of the Study

A limitation to the security assurance case extension is the effectiveness of the game-theoretic model. The extension as a concept is not dependent on the specific type of game model used, however the effectiveness of the extension at runtime is directly impacted by the game-theoretic model and how effective it is at identifying the optimal strategy to take and support the decision-making. Therefore, the extension is still limited by the uncertainty around the probabilities and payoffs that are the basis for the prediction of the optimal threat mitigation strategies. Additionally, in the future the model should be tested in more realistic contexts, where also the scalability of the concept can be assessed. This might show important results in regards to the type of game used and which one might be more effective than others.

Moreover, the interviews further showed that security assurance cases and how the security of a system is maintained in practice are very broad and complex areas. It was difficult for some participants to give insights in regards to how the incident response operates as well as how effective the extension might be in direct attack situations as this was not their domain. Additionally, the evaluation of the artifact in this thesis was evaluated with a rather small number of security experts. This might impact the generalisability of the results and could then also increase the risk of biased responses influenced by the respondents' individual backgrounds. However, a more exhaustive validation was not possible due to time constraints. Thus, more respondents as well as ones within the incident response or working with active attacks would have brought more insights into how the tool could be used in their area of work.

6.5 Future Research

Since the extension that is proposed here is only a concept, in future work it will need to be further developed with the feedback from the security experts. One aspect in the feedback was that the system impact needs to be clarified in order for the extension to be more operationally effective. This could possibly be implemented in connection with Bayesian Belief Networks that have been used as a mean to estimate the confidence in security assurance cases and could then potentially be used to evaluate system impacts during an attack as well. Moreover, an actual prototype needs to be created with the components detailed in relation to the quality claim that can then also be tested in more realistic scenarios or possibly contexts in order to give more tangible results on how effective this extension could be.

In addition to this, the security assurance case extension was intended to be a standalone product, therefore the thesis did not explore how to incorporate the results in practice and existing tools that might be used for designing security assurance cases. Thus, future research should focus on how the artifact can be incorporated and operate in practice with the tools typically used to work with security assurance cases.

Furthermore, the security experts involved in the evaluation of the model are very experienced within security assurance cases and work with them in different industries as well as domains. However, as discussed above, more experts that are directly involved with the incident response of security threats as well as active attacks will be needed to evaluate the extension in this context to gain insights on how it could work to support their processes as well as promote quick and effective decision-making.

In addition to this, in game theory, the players are assumed to be rational and intelligent. Therefore, it is assumed that players understand the consequences of their actions and are able to make intentional decisions. However, in reality that is not necessarily the case and therefore game theory overall might be not as effective, since players might not play in their best interest. Therefore, it could be interesting to further explore how the psychology of the attacker might impact the interactions with a system and how it can be accounted for in a game-theoretic model as well as the security assurance case.

7
Conclusion

In this thesis, the focus was to investigate how security assurance cases can be extended to include decision-support at runtime. Security assurance cases with the right quality assurance in place can inspire confidence in the system's security, however this also only applies to the claims in the case made at that time, therefore at design time. At runtime however, external factors might behave differently and therefore impact the system and its security which might not have been or could not be accounted for in the security assurance case.

Given that security is a dynamic concept, it is therefore important that the security assurance case is able to consider runtime behaviour and new information incoming during a threat scenario in order for a security team to be able to rely on the assurance case to make decisions. Consequently, the inclusion of game theory in the security assurance case enables dynamic changes in the case that reflect runtime threats and what strategies to take that keep the system secure, which can thus increase the confidence in the security assurance case and in turn the system's security. The current challenges in practice that have been identified can be addressed with a decision-support that is able to simulate runtime processes and guide the decision-making while also balancing the human and automated aspect in these decisions. Therefore, incorporating a game-theoretic model as evidence to support why certain strategies should be taken at runtime when faced with a certain attack can help guide the decision-making and actions of security teams and make security assurance cases more usable as well as easier to maintain at runtime. In addition to this, it could also serve as a support for validating claims at design time through test simulations with the security assurance case to identify potentially uncovered areas in the system's security.

The combination of security assurance cases and runtime adaptivity, also based on game theory, is a rather unexplored field with most of the literature focusing on dynamic evidence and evaluations or on specifically self-adaptive systems and instantiating or dynamically changing security assurance cases at runtime. However, the use of the proposed dynamic decision-support extends assurance cases to potentially become more useful during the decision-making and being able to benefit different types of systems as well as contexts. The evaluations highlighted some potential doubts in regards to the extension's scalability and how effective a security assurance case is for runtime decision-making, particularly in attack scenarios where there is little time, which need to be addressed in future work. This thesis presents an initial prototype that has to be further developed and tested in order

to gather more insights into these potential issues and identify more requirements under different and more realistic circumstances.

Bibliography

[1] O. Ochoa, J. Steinmann, and Y. Lischuk. "Towards Eliciting and Analyzing Security Requirements using Ontologies through Use Case Scenarios". In: *International Conference on Software Security and Assurance (ICSSA)*. 2018.

[2] M. Mohamad et al. "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)*. 2021, pp. 29–36.

[3] N. Mansourov and D. Campara. *System Assurance: Beyond Detecting Vulnerabilities*. 2011.

[4] K. Brown and Y. Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. 2008.

[5] M. Mohamad, J. Steghöfer, and R. Scandariato. "Security assurance cases - state of the art of an emerging approach". In: *Empirical Software Engineering* 26.70 (2021), pp. 472–486.

[6] S. Jahan et al. "MAPE-K/MAPE-SAC: An interaction framework for adaptive systems with security assurance cases". In: *Future Generation Computer Systems* 109 (2020), pp. 197–209.

[7] H. Lipson and C. Weinstock. "Evidence of Assurance: Laying the Foundation for a Credible Security Case". In: (2008). URL: `https://resources.sei.cmu.edu/asset_files/WhitePaper/2013_019_001_295685.pdf`.

[8] R. Alexander, R. Hawkins, and T. Kelly. "Security Assurance Cases: Motivation and the State of the Art". In: *High Integrity Systems Engineering Department of Computer Science University of York* 1.1 (2011).

[9] E. Yuan, N. Esfahani, and S. Malek. "A Systematic Survey of Self-Protecting Software Systems". In: *ACM Transactions on Autonomous and Adaptive Systems* 8.4 (2018), pp. 1–41.

[10] D. Snider et al. "Using Concept Maps to Introduce Software Security Assurance Cases". In: *ACQUISITION OF SOFTWARE-RELIANT CAPABILITIES* (2014), pp. 4–9.

[11] R. Calinescu et al. "Engineering Trustworthy Self-Adaptive Software with Dynamic Assurance Cases". In: *IEEE Transactions on Software Engineering* 44.11 (2018), pp. 1039–1069.

[12] J. Rushby. *The Interpretation and Evaluation of Assurance Cases*. 2015.

[13] C. Lin, W. Shen, and B. Cheng. “Measuring Confidence of Assurance Cases in Safety-Critical Domains”. In: *Proceedings of the 53rd Hawaii International Conference on System Sciences*. 2020, pp. 6355–6364.

[14] S. Nair et al. “An extended systematic literature review on provision of evidence for safety certification”. In: *Information and Software Technology* (2014), pp. 689–717.

[15] M. Ouedraogo et al. “A New Approach to Evaluating Security Assurance”. In: *2011 7th International Conference on Information Assurance and Security (IAS)*. 2011, pp. 215–221.

[16] N. Pham et al. “A near real-time system for security assurance assessment”. In: *Proceedings of the third International Conference on Interest monitoring and protection*. 2008, pp. 152–160.

[17] D. Weyns et al. “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. 2017, pp. 31–63.

[18] R. De Lemos et al. “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. 2017, pp. 3–30.

[19] D. Perez-Palacin and R. Mirandola. “Uncertainties in the modeling of self-adaptive systems: a taxonomy and an example of availability evaluation”. In: *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering, ICPE*. 2014, pp. 3–14.

[20] P. Nespoli et al. “Optimal countermeasures selection against cyber attacks: A comprehensive survey on reaction frameworks”. In: *IEEE Communications Surveys & Tutorials* 20.2 (2017), pp. 1361–1396.

[21] N. Ben-Asher and C. Gonzalez. “Effects of cyber security knowledge on attack detection”. In: *Computer in Human Behavior* 48 (2015), pp. 51–61.

[22] R. Lahcen et al. “Review and insight on the behavioral aspects of cybersecurity”. In: *Cybersecurity* 10.3 (2020), pp. 1–18.

[23] X. Liang and Y. Xiao. “Game Theory for Network Security”. In: *IEEE Communications Surveys Tutorials* 15.1 (2013), pp. 472–486.

[24] C. Kamhoua, F. Kiekintveld C.and Fang, and Q. Zhu. *Game theory and machine learning for cyber security*. John Wiley & Sons, 2021.

[25] P. Peskun. “Optimum Monte-Carlo sampling using Markov chains”. In: *Biometrika* 60.3 (1973), pp. 607–612.

[26] C. Do et al. “Game theory for cyber security and privacy”. In: *ACM Computing Surveys* 50.2 (2017), pp. 1–37.

[27] Eilon Solan and Nicolas Vieille. “Stochastic games”. In: *Proceedings of the National Academy of Sciences* 112.45 (2015), pp. 13743–13746.

[28] Jian Yao et al. “Solving Imperfect Information Poker Games Using Monte Carlo Search and POMDP Models”. In: *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*. 2020, pp. 1060–1065.

[29] P. Doshi, X. Qu, and A. Goodie. “Decision-Theoretic Planning in Multiagent Settings with Application to Behavioral Modeling”. In: *Plan, Activity, and Intent Recognition*. 2014, pp. 205–224.

[30] S. Edelkamp and S. Schrödl. *Heuristic Search*. 2012, pp. 3–46.

[31] Y. Chen et al. “Game Theoretic Markov Decision Processes for Optimal Decision Making in Social System”. In: *GlobalSIP 2014: Game Theory for Signal Processing and Communications*. 2014, pp. 268–272.

[32] M. Cheah et al. “Building an automotive security assurance case using systematic security evaluations”. In: *Computers Security* 77 (2018), pp. 360–379.

[33] D. Le. “Quality Trade-offs in Self-Protecting System”. In: *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*. 2015, pp. 152–156.

[34] M. Jalali, M. Siegel, and S. Madnick. “Decision-making and biases in cybersecurity capability development: Evidence from a simulation game experiment”. In: *The Journal of Strategic Information Systems* 28.1 (2019), pp. 66–82.

[35] K. Lye and J. Wing. “Game strategies in network security”. In: *International Journal of Information Security* 4.1 (2005), pp. 71–86.

[36] A. Patil, B. S, and N. Annigeri. “Applications of Game Theory for Cyber Security System: A Survey”. In: *International Journal of Applied Engineering Research* 13.17 (2018), pp. 12987–12990.

[37] Y. Wang, C. Lin, and K. Meng. “Analysis of Attack Actions for E-Commerce Based on Stochastic Game Nets Model”. In: *Journal of Computers* 4.6 (2009), pp. 461–468.

[38] J. McInerney et al. “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*. 2001.

[39] J. Zheng and A. Namin. “Defending SDN-based IOT networks against DDOS attacks using Markov Decision Process”. In: *2018 IEEE International Conference on Big Data*. 2018.

[40] A. Appelbaum et al. “Intelligent, Automated Red Team Emulation”. In: *Proceedings of the 32nd Annual Conference on Computer Security Applications*. 2016, pp. 363–373.

[41] A. Carstensen and J. Bernhard. “Design science research – a powerful tool for improving methods in engineering education research”. In: *European Journal of Engineering Education* 44.1-2 (2019), pp. 85–102.

[42] V. Braun and V. Clarke. “Using thematic analysis in psychology”. In: *Qualitative Research in Psychology* 3.2 (2006), pp. 77–101.

A

Appendix

A.1 Interview Guide

1. Background Questions
1.1. Years of experience with Security and Security Assurance
1.2. Their current role
1.3. Their main industry

2. Runtime Security and Decision-making
2.1. What kind of decisions have to usually be made at runtime regarding the security of the system? Is the human error in these decisions a common problem?
2.1.1. How would you make decisions at runtime when faced with a security threat when there are multiple strategies? What information are the decisions based on?
2.1.2. Who makes these decisions?
2.1.3. How often are you in situations where you have to make decisions at runtime with little to no support from design time documentation?

2.2. How do runtime changes or issues that arise at runtime affect the existing documentation for security assurance? How is that behaviour usually accounted for?
2.2.1. How do you communicate with people at designtime about the necessary changes that have to be made to documentation or issues encountered?
2.2.2. Who is responsible for updating and maintaining assurance, design time or runtime teams?

3. Design Time
3.1. What documentation do you have to aid decision-making at runtime? Do you communicate to runtime decision-makers in potential attack situations through e.g. is there a checklist or other documentation?
3.1.1. Is the documentation usually sufficient and covers multiple scenarios?

3.2. How do you anticipate runtime behavior when designing security and creating documentation to demonstrate security assurance?

3.3. What challenges are there to maintaining security assurance also at runtime?
3.3.1. Have you found any solutions to these challenges? What resources would be needed to realize them? What barriers were you facing for these solutions?

4. Transition between Design and Runtime

4.1. Since security assurance is created and assumptions are made at design time about runtime, do you do any fact checking of the design with the deployed system?

4.2. Is the designed documentation/assurance updated with for example threat analysis for the running system? Why or why not?
4.2.1 Who is typically responsible for this, design or runtime teams?

4.3. How much of an impact is it on the existing documentation and the general workflow/organization when it needs to be updated?

5. Implementations of dynamic decision-making and increase in confidence

5.1. What requirements would a dynamic decision-making system/support need to fulfill to be useful?

5.2. How much would the confidence in the system increase if you had a more dynamic decision-support?

A.2 Evaluation Questions

1. Closed Questions
1.1. How easy is the extension/decision-support to understand?
1 (Not easy at all) • 2 (Not very easy) • 3 (Somewhat easy) • 4 (Mostly easy) • 5 (Very easy)

1.2. How satisfied are you with the amount of information that is provided in the extension?
1 (Very dissatisfied) • 2 (Somewhat dissatisfied) • 3 (Neither satisfied nor dissatisfied) • 4 (Somewhat satisfied) • 5 (Very satisfied)

1.3. How satisfied are you with the type of information that is provided in the extension?
1 (Very dissatisfied) • 2 (Somewhat dissatisfied) • 3 (Neither satisfied nor dissatisfied) • 4 (Somewhat satisfied) • 5 (Very satisfied)

1.4. To what extent would you be able to confidently make runtime decisions with this extension?
1 (To no extent) • 2 (To little extent) • 3 (To some extent) • 4 (To a large extent) • 5 (To a great extent)

1.5. How effective do you think the extension would be in the team's decision-making and response to an attack?
1 (Not effective at all) • 2 (Not very effective) • 3 (Somewhat effective) • 4 (Mostly effective) • 5 (Very effective)

1.6. How realistic is the extension and simulation model and an accurate representation of an actual attack scenario?
1 (Not realistic at all) • 2 (Not very realistic) • 3 (Somewhat realistic) • 4 (Mostly realistic) • 5 (Very realistic)

1.7. To what extent does the extension as a whole increase the confidence you have in the system's security at runtime?
1 (To no extent) • 2 (To little extent) • 3 (To some extent) • 4 (To a large extent) • 5 (To a great extent)

1.8. How likely would you use the extension in your work or recommend it to others?
1 (Definitely not) • 2 (Probably not) • 3 (Possibly) • 4 (Probably) • 5 (Definitely)

2. Open-ended Questions
2.1. What is your general impression? What areas are easy or hard to understand?

2.2. Does the simulation accurately represent a realistic attack scenario?
• If not, what would need to be different/improved?

2.3. What might be some limitations of the extension in for example an actual attack scenario? What advantages can the extension provide in the attack scenario?

2.4. Are there any features that you think are missing in the extension that should be included to improve its effectiveness?

A.3 Coding Example

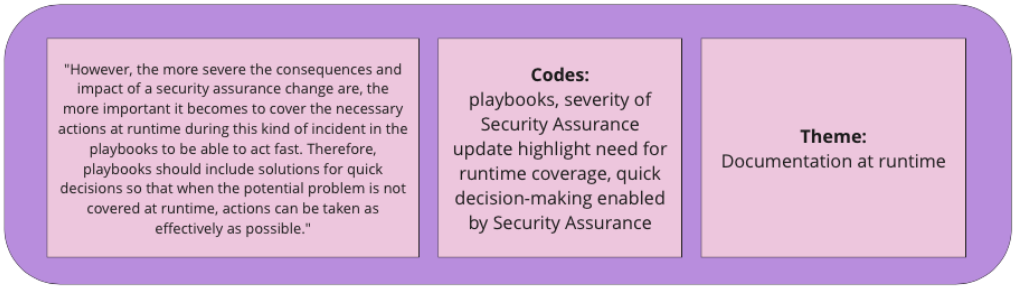

Figure A.1: Coding Example

Figure A.2: First Instance of Thematic Analysis of Interviews in Iteration 1

A.4 Security Assurance Case Extension

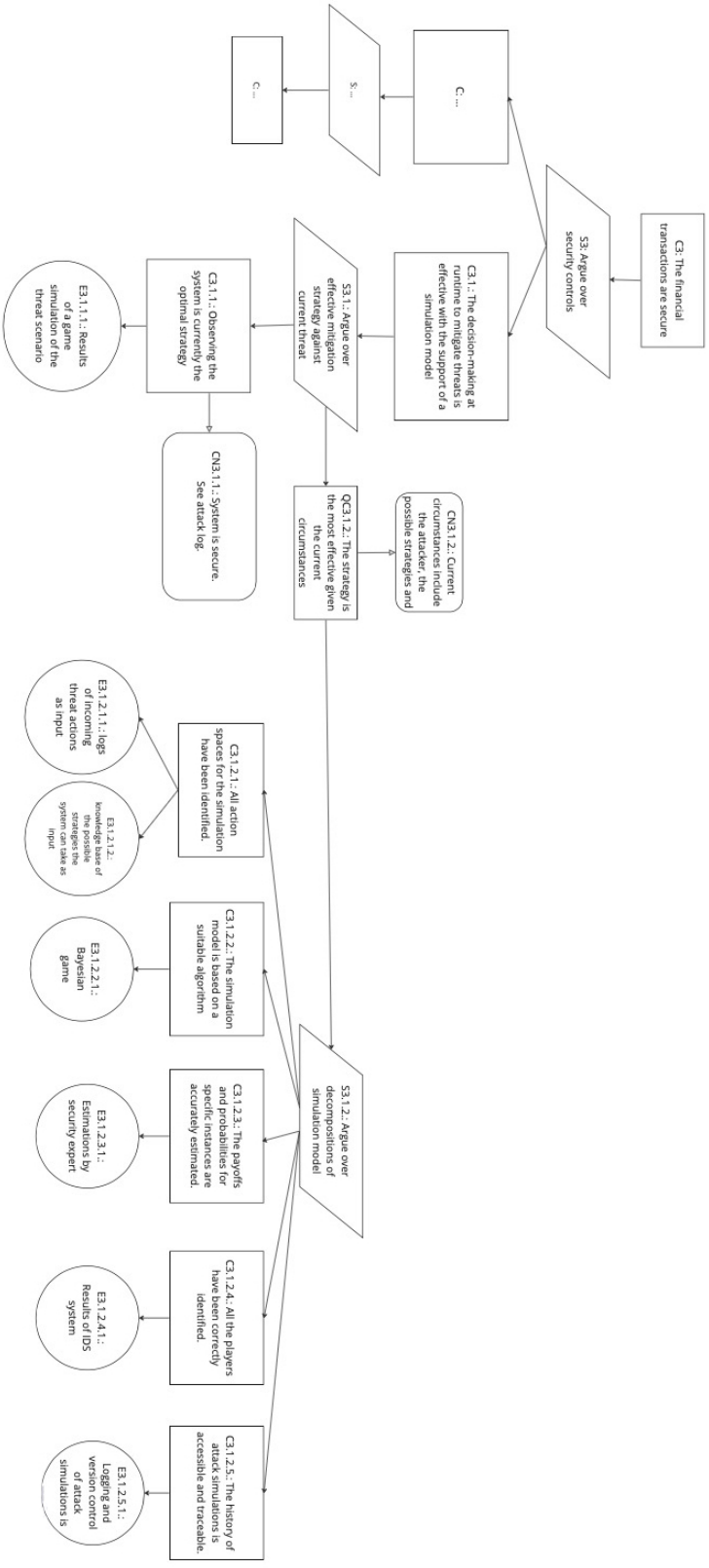

Figure A.3: Security Assurance Case Extension: System is Secure

A.5 Results of Validation

Table A.1: Central Tendency of Answers to Closed Questions in Artifact Evaluation

Question	Measure of	I1	I2	I3	I4	Mean	Mode	Median	Standard Deviation
1	Usability	4	2	4	3	3.25	4	4	0.829
2	Usability	3	4	5	3	3.75	3	4	0.829
3	Usability	4	2	3	4	3.25	4	4	0.829
4	Effectiveness	3	2	4	4	3.25	4	4	0.829
5	Effectiveness	4	3	5	4	4	4	4	0.707
6	Effectiveness	3	2	3	3	2.75	3	3	0.433
7	Confidence	4	3	4	4	3.75	4	4	0.433
8	Confidence	4	0	5	5	3.5	5	5	2.062
All	**Overall**					**3.438**	**4**	**4**	**1.059**