19th Annual Conference on Systems Engineering Research
Transdisciplinary nature of SE:
Impact on traditional and novel applications
March 24-26, 2022 - Norwegian University of Science and Technology

2022 Conference on Systems Engineering Research

A Conceptual Model-Based Systems Engineering Method for Creating Secure Cyber-Physical Systems

Martin H. Larsen[a]*, Gerrit Muller[a], Satyanarayana Kokkula[a]

[a]*University of South-Eastern Norway, Hasbergs vei 36, Kongsberg 3616, Norway*

Abstract
The Air Traffic Control industry is being increasingly exposed to rising levels of risk, as criminals and cyber-attackers look to exploit system vulnerabilities. Air Navigation Service Providers become more demanding regarding cybersecurity concerns in the products they acquire. Consequently, systems engineers need to consider cyber security concerns early in their system's development life cycle. Model-Based Systems Engineering methodologies are widely used to manage complex engineering projects in terms of system requirements, design, analysis, verification, and validation activities, leaving cyber security aspects aside. This paper presents a conceptual solution of a model-based security method that aims to enable systems engineers to perform threat modeling analysis of cyber-physical systems early and incorporate mitigation strategies into the system design, thereby reducing the cyber-physical system's overall security-related risks. Based on a real-life case study the method will be validated later during execution period from Jan. – May 2022.

Keywords: Cyber-Physical Systems; threat modeling; MBSE; Cyber Security;

1. Introduction

This research is conducted within the context of Jotron AS, which is an Air Traffic Control (ATC) technology company in Norway. The case company Jotron AS has hundreds of employees around the world that develop and produce complex systems i.e. ATC communication systems. Over the last ten years, Jotron has gone from being a subcontractor of systems in major ATC projects, to gradually becoming a supplier that delivers complete system-of-systems. This has resulted in the systems with increased complexity, and the departments in the organization have increased in size.

The critical issues of cyber security have attracted much attention in the aviation industry in recent years. The European aviation industry is being increasingly exposed to rising levels of risk, as criminals, hackers, and cyber-

* Corresponding author. Tel.: +0047 900 67 488;
E-mail address: martin.larsen623@gmail.com

attackers look to exploit vulnerabilities, cause chaos, and above all, financial gain at the expense of the aviation sector (EUROCONTROL, 2021). Continuous airspace operation is a key for passenger security and even national security. Integrating cyber security considerations in the design of modern systems is a significant challenge.

There is a need in the organization for a methodology to assure cybersecurity. Cybersecurity is the practice of protecting systems, networks, and programs from digital attacks. ATC systems worldwide are now modernizing, refreshing, or replacing aging infrastructures on ground and space, with the intent of substantially improving the capacity, safety, security, efficiency, and yield of aviation and outpace future demands. ATC systems are critical infrastructure for aviation safety, so therefore system reliability and security are extremely important. Fig. 1 illustrates an overview of security concepts and their relationships.

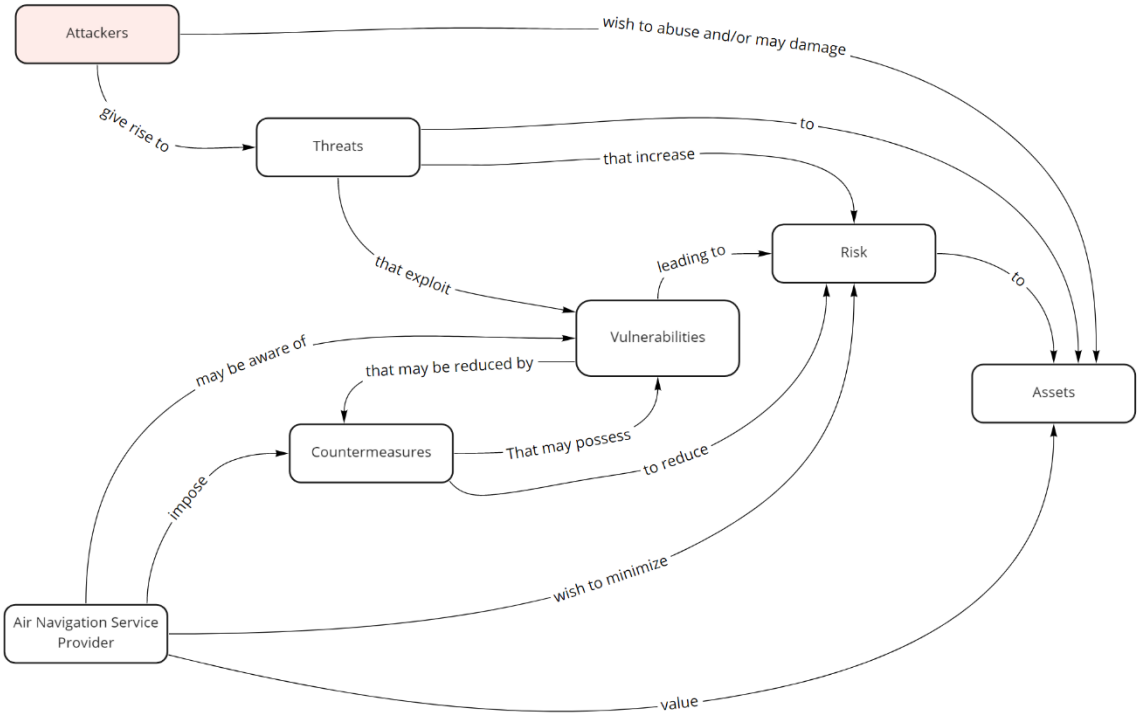

Fig. 1. Security Concepts and Relationships

Modern ATC systems have become IP-based with increased complexity, from the integration of several operationally independent systems. In this case transceiver system, recording system, Voice Communication System (VCS), Remote Radio Control (RRC) system, and secured network infrastructure have been brought together within the systems-of-systems umbrella. A system-of-systems is a collection of systems that were originally designed as stand-alone systems for specific and different purposes, but that have been brought together within the systems-of-systems umbrella to create a new capability needed for a particular mission. A high-level overview a network of ATC systems in its operational environments is shown in Fig. 2. Since modern ATC systems interact with each other and with their environment, these systems do not operate any longer in isolation. Thus, malicious actors may gain access to the infrastructure of these IP-based systems and, therefore, security for ATC becomes a topic of high relevance.

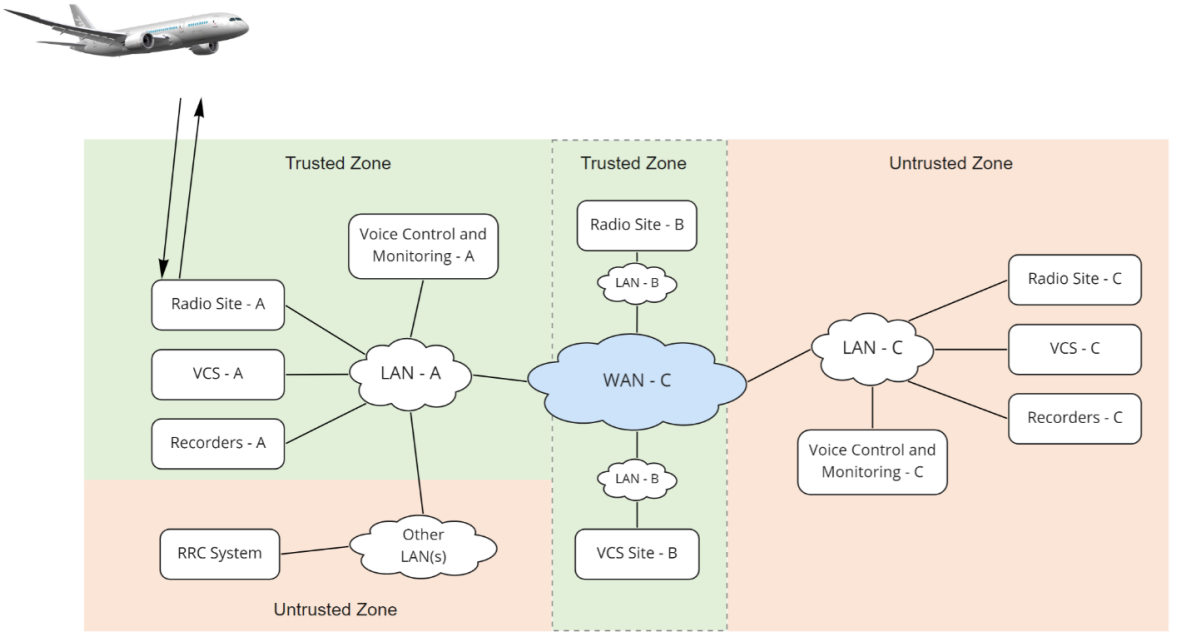

Fig. 2. ATC System Overview

Challenge: Modern ATC systems have become more complex due to systems have become cyber-physical systems and now depending upon the seamless integration of computational algorithms and various physical components.

The engineers in the ATC research and development department have an almost *ad-hoc* way of working with security analysis, due to the lack of a proper system development process. This often leads to increased development time, cost, reduced security, and inconsistency of the complex cyber-physical system development projects. On the other hand, the case company invests continuously in software tools that support software assurance. However, these tools are not sufficiently used concerning systems security. The current security analysis of system security in the development lifecycle is generally based on the engineer's knowledge, experience, and know-how from previous projects. Consequently, engineers use these security analyses either insufficiently or wrong in several projects.

To resolve the problem, this research aims to develop a method combining Systems Engineering and security engineering disciplines at an early stage of system development. The method developed in this research could apply to similar issues in other cyber-physical system industry cases.

Problem Statement

To address the challenge, the research focuses on answering the following research questions (RQ):

RQ1. How can cyber security risks be mitigated early in the system development process?

> *Rationale*. Mitigating cyber security risks early in the system-development process so that systems can become secure by design, in contrast to the common practice of adding security features later in the development process. Updating embedded ATC radio systems for fixing vulnerabilities at run time is in many cases difficult and leads to high costs.

RQ2. How to develop a method that integrates cyber security analysis activities into the Systems Engineering process in the organization for increased security?

> *Rationale*. One of the most important challenges the organization is trying to solve while creating new systems is how to achieve security-by-design. The system is treated as a secure system if the principles of confidentiality, integrity, and availability are guaranteed.

In the "as-is" situation in the organization, the system security engineering field includes a variety of methods and techniques for tackling security risks. However, they are disjointed from each other as well as from systems engineering. In the current dynamic cyber security threat environment, an integrated, agile methodology, and mindset are required to properly design, test, deploy, operate, and maintain secure systems in an ATC environment. Cyber security must be understood and integrated into all disciplines and through all phases of a system life cycle to meet requirements with acceptable levels of risk.

2. Research Methodology

The research methodology followed in this paper is based on case study. According to Robson (2002), "*case study is a strategy for doing research that involves an empirical investigation of a particular contemporary phenomenon within its context using multiple sources of evidence*". In the field of software engineering, Runeson et al. (2012) customized a guideline for case studies, including a theoretical framework, methods for data collection, and methods for data analysis and interpretation. Based on these guidelines, the case-of-interest in this research is the security method. Fig. 3 visualizes the research process.

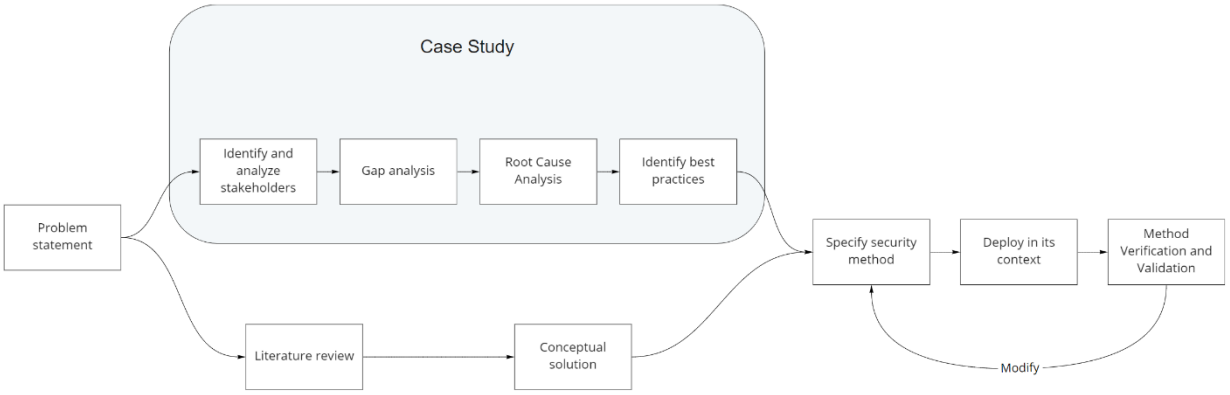

Fig. 3. Research Process

2.1. Literature review

The literature review covers the relevant state-of-the-art in Systems Engineering, Model Based Systems Engineering (MBSE), security requirements engineering, and security risk management fields. The literature review also identifies the specific knowledge areas that the research willuse. The evaluation of the state-of-the-art literature review aims to identify the limitations and potential needs in systems engineering and security areas, align concepts and techniques and select the core elements for the domain-specific language and the MBSE security method.

2.2. Conceptual solution

Based on the literature review, a conceptual solution is proposed regarding the use of the security method to assure system security. This research will adopt the system engineering problem-solving approach in this research from the identification of the problem to the implementation of a solution

2.3. Specify security method

Based on the case study findings and the conceptual solution, the first security method prototype shall be developed. The prototype creates an arena for continuous knowledge creation, verification, and validation.

2.4. Deploy in its context

Once a prototype of the method is developed, it is deployed in its context. So that they can gain hands-on experience to provide feedback and improvement suggestions.

2.5. Method verification and validation

Each iteration of the method prototype undergoes a verification and validation session. The version of the method prototype that is verified and validated to meet the research goal is released as the final solution.

During case study analysis, there will be three rounds of data collection. The objectives of the case study are to perform a stakeholder analysis, gap analysis, Root Cause Analysis (RCA), and to identify the best practices for the security method. The stakeholder analysis provides a means to make a qualified choice of participants for data collection sessions. The gap analysis is done to build awareness of the gaps between "as-is" and "to-be" situation in the context of security, which provides the ground for the RCA. The RCA will be used to analyze the root causes of the challenge. Table 1 provides details on data collection methods, participants, and objectives in the case study.

Table 1. Data Collection Methods, Participants, Experience, and Objective in the Case Study

	Unstructured interview#1	Focus group workshop #1	Unstructured interview#2
Participants	1 software engineer and 1 department manager	4 software engineers and 2 department managers	6 software engineers and 2 department managers
Experience	Combined experience of over 20 years in the ATC industry	Combined experience of over 75 years in the ATC industry.	Combined experience of over 90 years in the ATC industry.
Role	Key opinion leaders in software and firmware engineering in the radio department and manager. Identified from stakeholder analysis.	Key roles in software and firmware engineering in the department and manager. Identified from stakeholder analysis.	Key roles in software and firmware engineering in the department and manager. Identified from stakeholder analysis.
Objective	Identify and analyze stakeholders.	Gap analysis	RCA

The next phase is data collection sessions for verification and validation of the security method prototypes. The data collection sessions are also used to gather feedback that can improve the next iteration of the method. Table 2 describes data collection methods, participants, experience, and objectives.

Table 2. Method Verification and Validation

	Security Method Iteration #1	Security Method Iteration #2
Data collection methods	Security Method Presentation + Focus group workshop#2	Security Method Presentation + Semi-structured Interview#1
Participants	1 software engineer and 1 department manager	8 software engineers and 2 department managers
Experience	Combined experience of over 20 years in the ATC industry	Combined experience of over 50 years in the ATC industry
Role	Key opinion leaders in software and firmware engineering in the radio department and manager. Identified from on stakeholder analysis.	Key roles in software and firmware engineering in the radio department and manager. Identified from stakeholder analysis.
Objective	Method implementation and data collection from iteration#1	Verification and Validation

3. State-of-the-art

The field of cyber security is widely researched area for the software engineering discipline. There are several industry-acceptable methods to ensure secure software development throughout all phases of the development process, including the waterfall-based Microsoft Security Development Lifecycle (SDL) method (Howard & Lipner, 2006) and the NIST framework for Security Considerations (Kissel et al., 2008) as well as the Microsoft Security Development Lifecycle for Agile Development (Microsoft, 2012).

Researchers in their studies (Nguyen et al., 2017)(Papke, 2017) agree that there is a need to identify and mitigate security risks during the systems engineering lifecycle. According to (Stevens, 1998) "*Systems engineering is about effective solutions to problems, and manage the technical complexity of the resulting developments*". Nguyen et al., (2017) recommend that security concerns should be considered together with the business logic very early, which is crucial in engineering secure systems. Nejib et al. (2017) presented a methodology and processes framework to further expose and build an understanding of system security engineering artifacts and responsibilities for the system engineering community. Bayuk & Horowitz (2011) presented a methodology to identify classes of new reusable system security solutions and an architectural framework based on the reuse of the patterns of solutions.

Navas et al., (2020) presented a model-based engineering practice and technique enabling an effective co-engineering effort between cyber security and systems engineering. They found that in many cases the cyber security effort must occur separately from the main systems engineering effort, due to the high cyber security architectural analysis. Some of the benefits of using an MBSE approach compared to a traditional document-based approach are enhanced communications, improved quality, and enhanced knowledge transfer (Friedenthal et al., 2014). MBSE methodologies have been widely researched and used in a wide spectrum of industries (Vipavetz et al., 2016) (Malone et al., 2016). Still, there are not many attempts to standardize how the security analysis and threat-modeling analysis can be conducted in a model-based environment within the system engineering process. Mažeika (2021) described the opportunities of using MBSE for creating secure systems:

> "*The state-of-the-art analysis of related works has revealed that MBSE is the right application for incorporating security requirements engineering and security analysis activities into the system engineering process. The security aspect is crucial in designing a complex system; however, the most popular MBSE methodologies do not provide (or provide very limitedly) such capability."*

Hecht & Baum (2019) presented an automated Failure Mode and Effects Analysis (FMEA) generator using the SysML modeling language and described its application for reliability, safety, and cyber security for critical

infrastructure. They found that their analysis can be readily repeated throughout the design and can be used to identify weaknesses and take corrective actions to create a more resilient and robust system.

Geismann et al. (2018) described how secure software engineering practices can be integrated into an engineering process for cyber-physical systems, and how security requirements can be identified and specified at the systems engineering level, and how these security requirements can be addressed systematically by taking appropriate countermeasures during software engineering. They used attack-defense graphs as threat models for tracing security requirements to both application-level countermeasures and platform-level countermeasures. Thereby, they aimed to increase the overall security of systems because requirements, threats, and countermeasures are made explicit and are traceable across the whole development lifecycle. They proposed the integration of secure software engineering practices into the cyber-physical systems engineering process. In their research, they showed a promising and interesting technique for designing systems more secure. However, the paper lacks validation that their technique works in the real world. This is something the authors also concluded and is something they will try to do in the future.

Method A: MBSEsec.

Mažeika and Butleris (2020) presented how MBSE could be leveraged to mitigate security risks at an early stage of system development. Their paper analyzes various security-related techniques and then clarifies how these techniques can be represented in the SysML model and then further exploited with MBSE tools and introduce the MBSEsec method, which gives guidelines for the security analysis process, the SysML-based security profile, and recommendations on what security technique is needed at each security process phase. Mažeika and Butleris verified the MBSEsec method by creating an application case study and running an experiment where systems and security engineers evaluated the feasibility of their approach. The MBSEsec method presented in their paper consists of the SysML/UML-based profile, security process definition, and recommendations on how the specific security technique should be implemented. MBSEsec method covers phases starting from security requirements identification, continuing capturing assets and modeling threats and risks, and finally deciding security control objectives and appropriate controls.

Method B: ProCom

Saadatmand and Leveque (2012) presented a method to integrate security features in the model-driven engineering approach with ProCom. The approach utilizes existing model elements and implements runtime properties for security solutions. One advantage of this approach is that the original models are annotated and then transformed automatically into a security-aware system, which still conforms to the original meta-model.

Method C: CHASSIS

Raspotnig et al. (2013) presented a method that allows identifying both security and safety aspects and is based on UML notation. The CHASSIS method consists of three steps. The first two steps rely on creating and analyzing UML-based diagrams. The third steps suggest conducting results in a HAZard and Operability Analysis (HAZOP) table and in security requirements specification.

Method D: SysML-Sec

Apvrille and Roudier (2013) presented a model-driven engineering method that aims at fostering the collaboration between system designers and security experts in all phases of the systems development life cycle of embedded systems. SysML-Sec is based on SysML and provides customized SysML diagrams to describe security-related elements of the system.

Method E: SEED

Vasilevskaya et al. (2014) presened an approach for Security-Enhanced Embedded System Design (SEED). SEED is a model-oriented, domain-specific approach that explicitly focuseson separation of responsibilities and concerns. Their approach builds on the basic premise that models are viable means of communication for expert knowledge.

Table 3 is based on and merges the work from Mažeika (2021) and Geismann et al. (2020). The table presents security concepts with definitions, synonyms, and their occurrence in the analysis modeling methods ("+" means that the corresponding concept is used in the modeling approach, and "-" means that it is not relevant).

Table 3. Security concepts mapped to modeling methods

	Definition	Method						Synonyms
		A	**B**	**C**	**D**	**E**	**F**	
Asset	Elements that can be considered as a subject for security analysis	+	+	+	+	+	+	Software asset, system asset, data asset
Security constraint	A type of rule that captures a formal statement to define security laws, regulations, guidance, and policies	+	+	+	+	+	+	Security requiremen t, security goa
Security control	A safeguard or countermeasure prescribed for an information system, or an organization designed to protect the confidentiality, integrity, and availability of the asset's information and to meet a set of defined security requirements	+	+	-	+	-	+	Security activity, safeguard, countermea sure, security-related function
Security property	Property or constraint on a system asset that characterizes their security need	+	+	-	+	+	+	Informatio n-assurance property
Risk	A statement of the impact of an event on as-sets	+	+	+	+	-	+	Risk
Risk impact	The potential impact on the system due to a specific reason (availability, integrity, and confidentiality)	+	+	+	+	-	-	Harm, consequenc e, security impact property
Probability	The likelihood of risk occurrence	+	+	-	-	-	+	Likelihood
Vulnerability	An internal fault that enables an external fault to harm the system	+	+	+	-	+	+	Weakness

Attacker	Someone or something attacking for altering the system's functionality or performance or accessing confidential information	+	-	+	+	+	+	Intruder
Threat	Potential attack that targets system assets and that may lead to harm to the assets. An action carried out to harm the system	+	+	+	+	+	+	Attack

Table 4 presents an overview of techniques for system security analysis and which security-related techniques overlap between analysis modeling methods. (Table 4 is based on and merges the work from Mažeika (2021) and Geismann et al. (2020)).

Table 4. Security risk analysis methods mapped to modeling methods

	Method						**Synonyms**
	A	**B**	**C**	**D**	**E**	**F**	
Security Risk Definition	+	+	+	+	-	+	Identifies and summarizes risks, risk impact, probability
Misuse Cases	+	+	+	+	+	-	Identifies threats and attackers
Misuse Case Sequence	+	+	-	+	-	+	Defines the attack sequence during an intrusion
HAZOP	-	-	+	-	-	-	Summarizes risk and security requirements-related data
Threat Scenario	+	+	-	+	+	+	Describes attack actions
Dolev–Yao Attacker Model	+	-	-	-	-	+	Formally defines potential actions by an attacker

4. Conceptual Solution

This section presents a conceptual solution to a model-based security method that aims to enable systems engineers to perform threat-modeling analysis of cyber-physical systems early and incorporate mitigation strategies into the system design, thereby reducing the system's overall security-related risks. Based on the literature review a conceptual solution of the method was sketched. Fig. 4 shows the phases and underlying security techniques of the conceptual solution.

Phase 1. Identify Security Requirements. The first phase of the method captures the security requirements as part of the functional and non-functional requirements. The security requirement refinement can be linked with Use Case diagrams.

Phase 2. Capture and Allocate Assets. The second phase is dedicated to defining the objects that the organization should secure and allocating them to the assets.

Phase 3. Model Threats and Risks. The third phase consists of behavioral and structural security specifications. For the behavioral risk and threat, definition Use Case diagrams are used for identifying Misuse Cases and the Attack Scenarios captured in Activity diagrams.

Phase 4. Decide Objectives and Controls. The fourth phase defines security control objectives and control

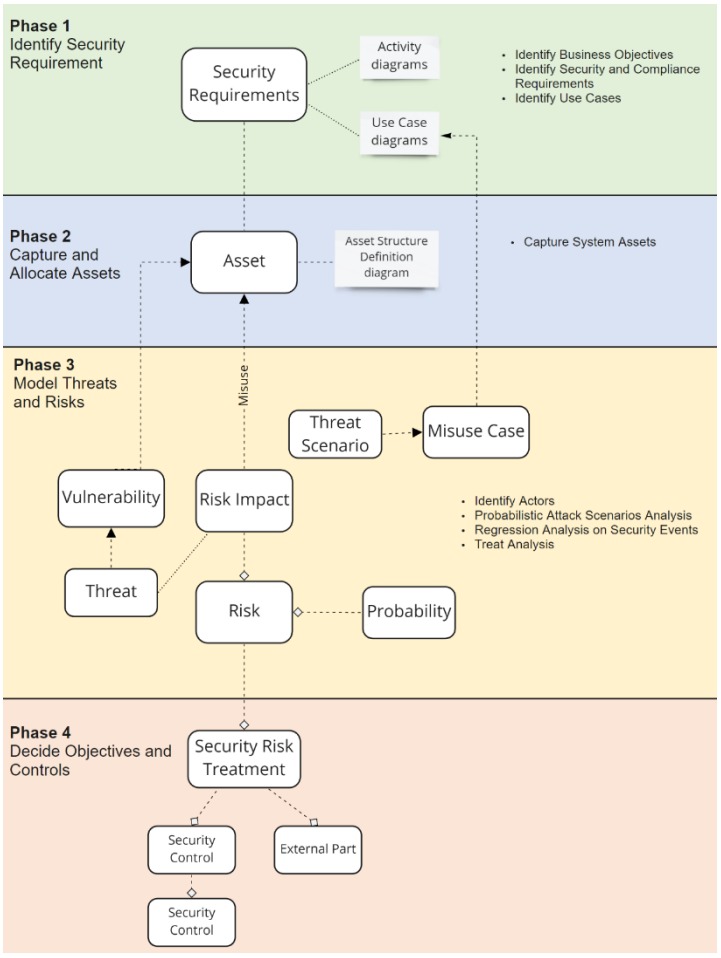

Fig. 4. Conceptual Solution MBSE Method for Creating Secure Cyber-Physical Systems

5. Discussion

A challenge when developing secure systems is to consider cyber security constraints while defining the best possible architecture. Systems engineering emphasizes analyzing the problem before jumping straight to the solution, as a means to develop systems effectively contributing to achieving stakeholders needs. To prevent threats from taking advantage of system flaws, systems engineers can use threat-modeling methods to inform defensive measures.

The conceptual method we have presented in this paper can potentially support collaboration and communication between several developers, in order to create more secure systems. Through a common and shared comprehension of the security operational context, the applicable security requirements, and the security constraints. Based on a real-life case study the method will be validated later during execution period from Jan. – May 2022. In our future reserarch we will answer RQ1 and RQ2 in the context of the case company.

5.1. RQ1. How can cyber security risks be mitigated early in the system development process?

The the real-life case study the security method will be used by 8 software engineers and 2 department managers over 5 months. These stakeholders are divided into two groups that work with incremental improvement of two

independent systems. The group working in the ATC radio department and the second group working in the recording system department. With a proper security-driven system development method, we believe that cyber security threats can be effectively identified and mitigated early in the system development process. Using threat modeling to consider security requirements may lead to proactive architectural decisions in the system architecting and design phase that can potentially help reduce threats from the start.

5.2. RQ2. How to develop a method that integrates cyber security analysis activities into the systems engineering process in the organization for increased security?

Introducing a method with the belief that that will improve security will probably have the opposite effect (false sense of security). Embeding specific design steps, based on previous experiences, and then validated, may improve security.

To evaluate the feasibility of the method, we need to ask the relevant stakeholders to answer questions related to their experience, work principles, and the security method itself. We will also ask if the participants could compare their efficiency when they move from document-based system engineering to model-based system engineering.

5.3. Expected results

There are reasons to believe that the final solution of the model-based security method can support cross-functional (systems, software, and security) teams to perform security analysis in parallel to the systems engineering process at an early stage of system creation. The model-based method is also expected to help the engineers to:

- eliciting and specifying security requirements
- identifying part of the system that could be vulnerable
- summarizing vulnerabilities, risks, and their impact
- define security controls and countermeasures

6. Conclusion

Almost all software systems today face a variety of threats, and the number of threats grows as technology changes. The ATC industry is being increasingly exposed to rising levels of risk, as attackers look to exploit system vulnerabilities. Air Navigation Service Providers become more demanding regarding cybersecurity concerns in the products they acquire. Consequently, systems engineers need to consider cyber security concerns early in their system’s development life cycle. In the current dynamic cyber security threat environment, an integrated, agile methodology and mindset are required to properly design, test, deploy, operate, and maintain secure systems in an ATC environment. Cyber security must be understood and integrated into all disciplines and through all phases of a system life cycle to meet requirements with acceptable levels of risk.

MBSE methodologies are widely used to manage complex engineering projects in terms of system requirements, design, analysis, verification, and validation activities, leaving cyber security aspects aside. One way to increase the security in a complex system creation is to remove silos between systems engineering and security teams and tackle security risks during the systems engineering lifecycle. Risk identification and mitigation are the most effective and maximize the return on investment if it is integrated into the design process and applied in the early stages.

This paper presents a conceptual solution of a model-based security method that aims to enable systems engineers to perform threat-modeling analysis of the cyber-physical systems early and incorporate mitigation strategies into the system design, thereby reducing the system's overall security-related risks.

References

1. EUROCONTROL. EUROCONTROL EATM-CERT Services Aviation under attack : Faced with a rising tide of cybercrime , is our industry resilient enough to cope ? 2021;(July).
2. Robson C. *Real World Research: A Resource for Social Scientists and Practitioner-Researchers*. Wiley-Blackwell; 2002.
3. Runeson P, Host M, Rainer A, Regnell B. *Case Study Research in Software Engineering: Guidelines and Examples*. 1. Aufl. Wiley; 2012.
4. Howard M, Lipner S. *The Security Development Lifecycle*. Vol 8. Microsoft Press Redmond; 2006.
5. Kissel R, Stine K, Scholl M, et al. Security Considerations in the System Development Life Cycle. Published online 2008. doi:10.6028/NIST.SP.800-64R2
6. Security Development Lifecycle for Agile Development | Microsoft Docs. Accessed December 3, 2021. https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee790621(v=msdn.10)
7. Nguyen PH, Ali S, Yue T. Model-based security engineering for cyber-physical systems: A systematic mapping study. *Information and Software Technology*. 2017;83:116-135. doi:10.1016/j.infsof.2016.11.004
8. Papke BL. Enabling design of agile security in the IOT with MBSE. In: *2017 12th System of Systems Engineering Conference, SoSE 2017*. Institute of Electrical and Electronics Engineers Inc.; 2017. doi:10.1109/SYSOSE.2017.7994938
9. Stevens R. *Systems Engineering: Coping with Complexity*. Pearson Education; 1998.
10. Nejib MP, Grumman N, Beyer D, Martin L, Yakabovicz ME. *Systems Security Engineering: What Every System Engineer Needs to Know*.; 2017.
11. Bayuk JL, Horowitz BM. An architectural systems engineering methodology for addressing cyber security. *Systems Engineering*. 2011;14(3):294-304.
12. Navas J, Com; JN, Voirin JL, Paul S, Bonnet S. *Towards a Model-Based Approach to Systems and Cybersecurity Co-Engineering in a Product Line Context*.; 2020.
13. Friedenthal S, Moore A, Steiner R. *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann; 2014.
14. Vipavetz K, Shull TA, Infeld S, Price J. Interface Management for a NASA Flight Project using Model-Based Systems Engineering (MBSE). In: *INCOSE International Symposium*. Vol 26. Wiley Online Library; 2016:1129-1144.
15. Malone R, Friedland B, Herrold J, Fogarty D. Insights from large scale model based systems engineering at Boeing. In: *INCOSE International Symposium*. Vol 26. Wiley Online Library; 2016:542-555.
16. Mažeika D. Model-based systems engineering method for creating secure systems. Published online 2021. https://ktu.edu.
17. Hecht M, Baum D. Use of SysML for the creation of FMEAs for Reliability, Safety, and Cybersecurity for Critical Infrastructure. *INCOSE International Symposium*. 2019;29(1):145-158. doi:10.1002/j.2334-5837.2019.00594.x
18. Geismann J, Gerking C, Bodden E. Towards ensuring security by design in cyber-physical systems engineering processes. In: *ACM International Conference Proceeding Series*. Association for Computing Machinery; 2018:123-127. doi:10.1145/3202710.3203159
19. Mažeika D, Butleris R. MBSEsec: Model-based systems engineering method for creating secure systems. *Applied Sciences (Switzerland)*. 2020;10(7). doi:10.3390/app10072574
20. Saadatmand M, Leveque T. Modeling security aspects in distributed real-time component-based embedded systems. In: *2012 Ninth International Conference on Information Technology-New Generations*. IEEE; 2012:437-444.
21. Raspotnig C, Katta V, Karpati P, Opdahl AL. Enhancing CHASSIS: a method for combining safety and security. In: *2013 International Conference on Availability, Reliability and Security*. IEEE; 2013:766-773.
22. Apvrille L, Roudier Y. SysML-Sec: A SysML environment for the design and development of secure embedded systems. *APCOSEC, Asia-Pacific Council on Systems Engineering*. Published online 2013:8-11.
23. Vasilevskaya M, Gunawan LA, Nadjm-Tehrani S, Herrmann P. Integrating security mechanisms into embedded systems by domain-specific modelling. *Security and Communication Networks*. 2014;7(12):2815-2832.
24. Geismann J, Bodden E. A systematic literature review of model-driven security engineering for cyber–physical systems. *Journal of Systems and Software*. 2020;169. doi:10.1016/j.jss.2020.110697