Enabling Airworthiness Security by a Holistic Security Engineering Process at various Aircraft Design Levels

Marvin Blecken[1], Hartmut Hintze[1], Ralf God[1]

[1]Hamburg University of Technology (TUHH), Institute of Aircraft Cabin Systems (FKS), Hamburg, Germany

Abstract

Since the publication of the EASA certification specification CS-25.1319 in 2020, cyber security of aircraft and aircraft systems (airworthiness security) became mandatory for aircraft certification and must be considered in the aircraft and aircraft systems development even before the detailed design and implementation. For protection of aircraft from cyberattacks by Intentional Unauthorized Electronic Interaction (IUEI), security risk analyses are now embedded in the early phases of aircraft and aircraft systems development and security information has to be provided to subsequent phases. But this also requires, that processes and methods employed at the detailed design and implementation level are compatible to the processes and methods from the level above. For this reason, regulations and corresponding standards are analyzed in this paper to define requirements for a comprehensive and holistic security engineering approach that ensures traceability and consistency along the various development levels in the context of model-based systems, software, and security engineering.

Keywords: Airworthiness Security, CS-25.1319, Security Engineering Process, Model Based System Security Engineering

1. Introduction

The development and certification of aircraft and aircraft systems are subject to regulations, which are published by the regulatory authorities, such as the European Union Aviation Safety Agency (EASA) in the European region. Until 2020, the certification specifications published by EASA were primarily concerned with matters of safety and reliability with the objective of ensuring safety of aircraft, passengers and the crew which comprises cabin and flight crew. Given the traditional concept of physically and logically segregating systems, especially safety-critical systems, the topic of cyber security was notably absent from the certification specifications but handled with special conditions set by responsible agencies for specific aircraft types such as the Airbus A380 or Boeing B787 [1]. However, the concept of logical and physical segregation is not feasible when it comes to optimizing existing or developing novel operational and business processes for the airline, as well as introducing digital services for passengers. These processes and services are characterized by digitalization and the communication of information enabled by connected systems. The necessity of interconnected and communicative systems and components is made evident by the example of condition monitoring and preventive and predictive maintenance of systems and components such as seat actuators in business class seats. To realize this example, actuator data is collected and communicated to appropriate backup solutions, which provide the computational resources needed for analysis and monitoring of the data. Furthermore, additional external interfaces permit the transfer of these data sets into the broader ecosystem within which the aircraft operates. This enables component manufacturers to leverage the data for further processing. But the interconnection of systems and their constituent components with external entities also allows for access to the systems, thereby increasing the susceptibility to cyberattacks by Intentional Unauthorized Electronic Interaction (IUEI). In order to ensure uniformity in the security processes defined by the special conditions for specific aircraft types and with the aim of protecting the aircraft from IUEI that may

affect the safety, EASA published the certification specification (CS) CS-25.1319 [2] in 2020. This obligates original equipment manufacturers (OEMs) and system and equipment suppliers to identify, assess, and mitigate security risks [2]. In consequence of CS-25.1319, security risk analyses have become a fundamental aspect of aircraft systems development. Moreover, the traditional approach of assigning security to the software lifecycle and treating security only in the detailed design and implementation, is no longer sufficient. Instead, a security risk analysis must be embedded in the development cycle of aircraft systems. In the context of model-based security engineering, the security risk analysis is integrated in the system model to enhance consistency and traceability. However, model-based approaches for the security risk analysis are neither connected to activities and models at the item level, nor are model-based methods for the detailed design, verification and validation of secure software/hardware items investigated and tailored with respect to the aviation-specific security standards. Subsequently, the current development of safe and secure aircraft systems lacks of a comprehensive and holistic security engineering approach which is necessary to ensure compatibility between the processes and methods employed at the different development levels as well as consistency and traceability along the various levels of development. This compatibility enables the communication and reuse of security information, such as threat scenarios, along the different development levels. As a result, the efficiency of the development of aircraft systems can be increased. Furthermore, the security of the system under development can be improved by preventing the inadvertent introduction of vulnerabilities due to a lack of communication and misunderstanding of security information and design flaws.

In light of the aforementioned context, a more comprehensive and holistic security engineering approach is envisioned in this paper. To this end, the regulatory context, including development and security standards introduced by the new certification specification is described and analyzed with a particular focus on the system level, the detailed design and implementation level, hereafter called as item level, and the transition between both levels. Based on this analysis, requirements for the holistic security engineering approach, especially for the transition between the system and item level, are defined. Further requirements are deduced in the context of model-based systems, software, and security engineering by considering a method for the security risk analysis and a method labeled UMLsec [3] which is applied for the verification of security requirements in the software design.

The following sections of this paper are structured as follows. In Section 2, the regulatory context to develop secure aircraft systems is described. Aside from the certification specifications, a detailed overview of the acceptable means of compliance for CS-25.1309 and CS-25.1319 (*i.e.* safety and security) is given. Subsequently, the state of the art and related work is presented in Section 3. Considering the regulatory context and the results of Section 3, requirements for a holistic security engineering approach with the focus on the transition between the system level and the item level are defined in Section 4. In Section 5, further requirements are derived by analyzing the transition between the system level an item level with consideration of model-based methods, i.e. model-based risk analysis (system level) and model-based verification (item level). Finally, a conclusion and an outlook for future work is given in Section 6.

2. Regulatory context

The development of aircraft systems occurs within a highly regulated environment. In this context, civil aviation authorities publish regulations based on national or transnational regulations which must be demonstrated to be adhered to during certification. This paper focuses on the certification specifications and related acceptable means of compliance published by EASA for aviation in Europe. Figure 1 provides an overview of the regulatory context relevant to the development of secure aircraft systems.

2.1 Certification Specifications

For the development of aircraft systems, CS-25.1309 has long been the most relevant certification specification. This certification specification outlines the requirements for the development process, with a particular focus on functional safety and reliability [2]. In light of the ever-evolving characteristics of modern aircraft systems, particularly the increasing reliance on new information and communication technologies to facilitate connectivity between systems, it is evident that the

number of risks has emerged. These risks extend beyond those inherent to safety and reliability, and encompass external threats that may arise from malicious interactions with the aircraft or aircraft systems. Consequently, in order to protect the aircraft, passengers and the crew from cyber threats and potential risks, it is inevitable that safety and security are considered complementary within the development process. This is underpinned by introduction of the new CS-25.1319 which demands the management of security risks including the identification, assessment and mitigation of security risks.

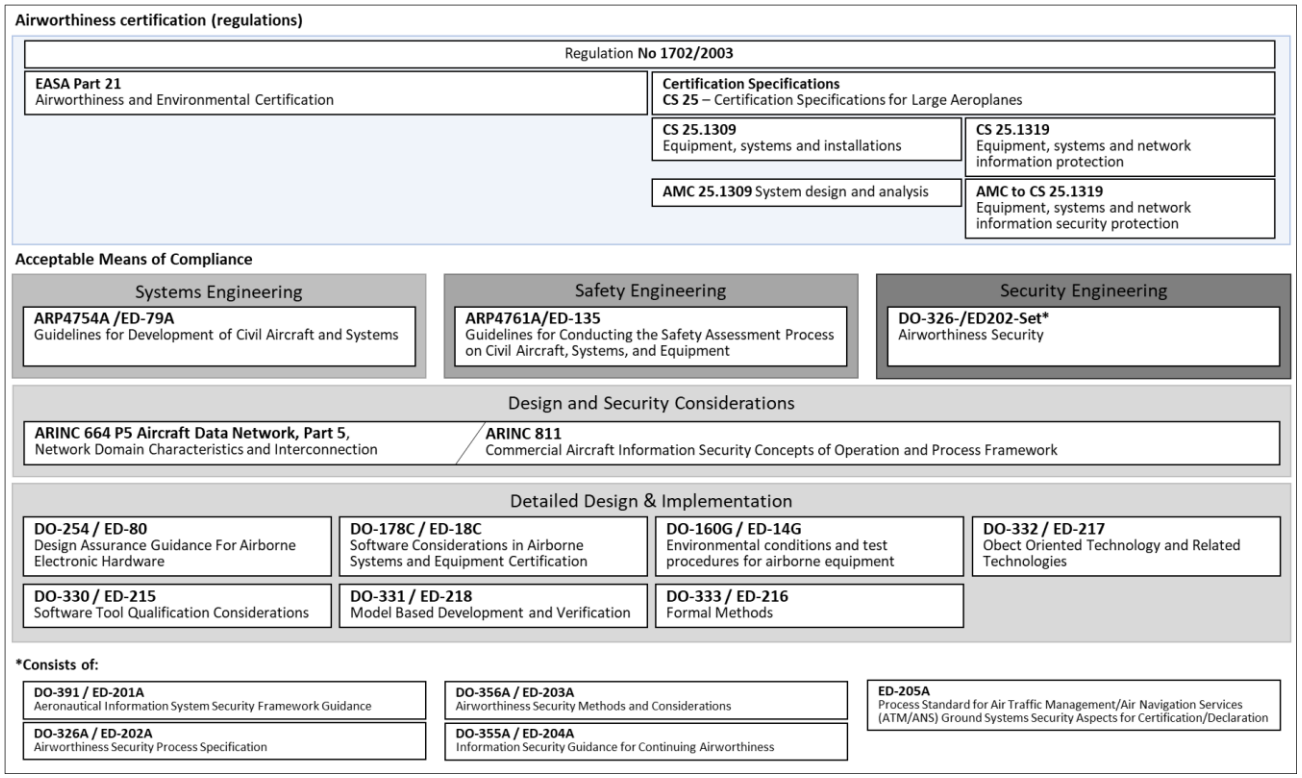

Figure 1: Regulatory Context based on [1] and [4]

2.2 Acceptable Means of Compliance

In addition to the certification regulations, EASA issues Acceptable Means of Compliance (AMCs). AMCs may reference aviation-related standards that provide processes and methods that can be applied to demonstrate compliance with the requirements defined in the certification specification. In the context of CS-25.1309, ARP4754A [5] (Aircraft and System Development), ARP4761A [6] (Safety Assessment), DO-178C [7] (Software Development), and DO-254 [8] (Hardware Development), among others, are considered as AMC. With regard to CS-25.1319, EASA has defined ED-202A [9], ED-203A [10] and ED-204A [11], which are part of the DO-326/ED-202 set [1], as an AMC, see Figure 1. In addition to these security-specific standards, ARP4754A and DO-178C are of particular interest for the development of a consistent security engineering approach whereas hardware development, whereas DO-254 is not considered in this paper.

The ARP4754A standard, published by the Society of Automotive Engineers (SAE), defines a development process for aircraft systems [5]. This process is based on the so-called V-model [12], which has been adapted to the specific needs and characteristics of the aviation industry. One of these characteristics is the large scale of aircraft systems which is why different levels of abstraction, also referred to as design or development levels, structure the development process. In the ARP4754A standard, three development levels are introduced, namely the aircraft, system and item level. Additionally, the standard describes interfaces with other standards, including ARP471A, DO-178C, and DO-254. The detailed design and implementation, i.e. the software development, is allocated to the item level. To this end, DO-178C defines software lifecycle processes, such as a software requirements process, a software design process, a software coding process and a software verification process [7]. The rigor of the software development is based on the objectives defined in DO-178C and depends on the Design Assurance Level (DAL) specified within the safety engineering process. There are other standards that complement DO-178C and DO-254 and cover topics such as tool qualification [13], model-based methods for software development and verification [14] or formal methods [15]. These are not within the scope of this paper although the security engineering approach

presented in this paper employs model-based methods.

In accordance with the requirement concerning the security risk management defined in CS-25.1319, the ED-DO-set, which has been jointly developed by the European Organization for Civil Aviation Equipment (EUROCAE) and the Radio Technical Commission for Aeronautics (RTCA), is indicated as an AMC. This comprehensive set encompasses a multitude of standards, which define processes, methods, and guidelines pertaining to both airworthiness security and continuing airworthiness security as illustrated in Figure 1. Airworthiness security is defined as “the protection of the airworthiness of an aircraft from IUEI” [10] and continuing airworthiness security includes the continuous monitoring of security vulnerabilities in the operation phase. The focus of this paper is on (initial) airworthiness security, which is why continuing airworthiness security is not considered.

ED-202A [9] describes an airworthiness security process that is conducted in parallel with the system and safety engineering process and encompasses process activities for certification, security risk analysis and security development [9]. These activities are mainly associated with the aircraft level and the system level, as exemplified by the Preliminary Aircraft Security Risk Assessment (PASRA) or the Preliminary System Security Risk Assessment (PSSRA). Furthermore, the ED-202A outlines the fundamental concepts that are used within the defined airworthiness security process. Among others, one concept is that of security assurance which is analogous to design assurance as defined in ARP4754A/DO-178C and comprises actions required to provide confidence and evidence that a system satisfies specified security objectives and security requirements [10].

The ED-203A [10] provides a systematic approach and methods to implement the airworthiness security process outlined in ED-202A, accompanied by illustrative examples of its application. Therefore, in ED-203A methods with respect to security scope definition, threat condition identification and effect evaluation, threat scenario identification and risk assessment for conducting a security risk analysis are provided. Moreover, in the context of security assurance, the ED-203A introduces the Security Assurance Level (SAL) and describes security assurance objectives. The extent of security objectives to be satisfied depends on the SAL which is determined in the security risk analysis. A distinction is made between security-specific assurance objectives and security development objectives. Both assurance objectives are not only intended for aircraft and system level but also require processes and actions at item level.

In addition to the aforementioned ED-standards, ARINC 664P5 [16] and ARINC 811 [17] define a security domain model which allocates system functions based on their criticality and operating context to different security domains and provide measures and guidelines to secure the transition between the different security domains.

3. State of the Art and Related Work

Based on the regulatory context described in Section 2, the so-called 2-V-model consisting of a System Engineering Process (SEP) “vee” and a Safety Engineering Process (SafEP) “vee” (*i.e.* two vee) [18] was extended by a third Security Engineering Process (SecEP) “vee” in [19] as shown in Figure 2. In the so called 3-V-model (*i.e.* three vee), the SEP, SafEP, and SecEP are arranged in parallel and are interconnected, thereby emphasizing the interaction and joint application of the processes. In particular, in [19] the interconnection between the SEP and the SecEP was considered and discussed in detail. Security Context Parameters (SCP) are proposed as a strategy for standardized communication between the processes at a certain development level, as illustrated in Figure 2. However, the exchange of information and communication between different development levels within a process or between the SEP and SecEP is not considered in [19].

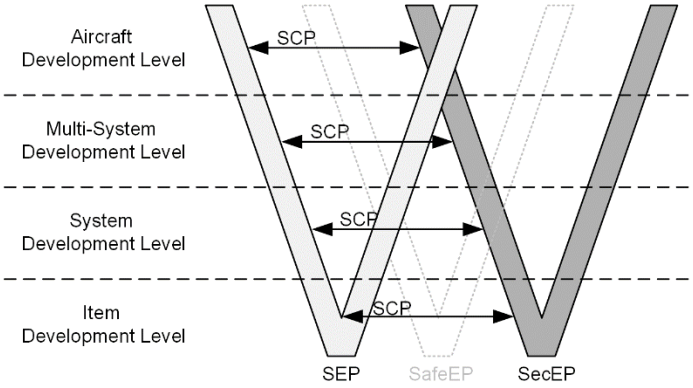

Figure 2: 3-V-Model and Interconnection between SEP and SecEP according to [19]

From a methodical point of view, different approaches exist, which deal with providing a more comprehensive and coherent security engineering approach especially in the context of model-based engineering. In [20] a goal-driven security requirement engineering method is combined with a model-based security engineering approach. In particular, UMLsec is integrated to the secure Tropos method [21] to provide a coherent and comprehensive security engineering approach while making use of the strength of both methods. By applying this combined methodology, security can be considered throughout all stages, from the elicitation of (security) requirements to the deployment, in the software development [20]. However, according to the regulations and associated acceptable means of compliance in aviation, security requirements are elicited by conducting a security risk analysis at the aircraft and system level. Furthermore, in a model-based context, the Systems Modeling Language (SysML) [22] is commonly used at the aircraft and system level to model (security) requirements as well as the aircraft and system design. Although the framework in [20] fosters holistic security engineering, the necessity to connect the security risk analysis conducted at higher development levels with security design and analysis at detailed design and implementation level with consideration of aviation-specific development standards is not addressed.

Matulevičius [23] focuses on secure system development and security modeling. In [23], different languages and methods for security modeling with emphasis on security risk modeling are discussed and also an approach to link multiple modeling perspectives by application of model-driven technologies, such as model transformation, is presented. In addition to the application of modeling for security risk management, the usage of security models to validate security requirements and to test the secure software design with a particular focus on the principle of role-based access control is discussed considering the UMLsec and SecureUML [24] methods. Both methods are also linked by means of transformation rules [23]. Even though the principle of a consistent security engineering process along different development phases is discussed, it is not explicitly shown, how the secure software design using the UMLsec or SecureUML approach can be linked to the security risk models.

BriefCASE [25] is another model-based systems engineering environment which is developed within the Cyber Assured Systems Engineering (CASE) program of the Defense Advanced Research Projects Agency (DARPA). The BriefCASE environment encompasses various tools for the design, analysis, and verification of cyber-resilient avionic components. Starting from system architectures modeled using the Architecture Analysis and Design Language (AADL) [26], security requirements can be suggested by using analysis tools that perform a vulnerability analysis. Based on the suggested security requirements, model transformations are used to integrate formally specified security measures, such as filters or monitors, in the software architecture. The formal specifications can then be used for formal verification to proof that the security-enhanced architecture satisfies the security requirements. Finally, a tool is included which enables the automatically generation of software code from the architecture [25]. The approach demonstrates the continuity of model-based development of cyber-resilient software through the use of model transformations, exemplified by the field of avionics. However, it is based on the AADL. The objective of this paper is to propose a comprehensive security engineering approach within the context of model-based development which is based on the Systems Modeling Language, a modeling language commonly employed in the aviation industry. Moreover, a security risk analysis as required according to [2] is not considered in the BriefCASE environment.

In [27], [28], a methodology for designing safe and secure embedded systems with the SysML using the SysML-sec environment [29] is described. Within this approach, formal verification can be performed based on SysML-sec models by transforming these models to a formal specification and using a dedicated tool to prove security properties [27], [28]. In this methodology, security analysis, i.e. verification of security requirements, is conducted based on the SysML whereas especially at the item level modeling languages specifically designed for software development, such as the Unified Modeling Language (UML) [30] are used.

It is important to emphasize that so far there is no comprehensive methodology that connects security risk analysis models to lower-level model-based security analysis, using the prevalent modeling languages SysML and UML and having the security aspects incorporated in the system model.

4. Requirements for a Holistic Security Engineering Approach

This paper introduces and utilizes the 3-V-model, as described in Section 3, as a foundation. In contrast to the horizontal exchange of information between the SEP and SecEP at one development level as illustrated in Figure 2, a holistic security development approach necessitates a transversal and vertical exchange of information along the various levels of development, as illustrated in Figure 3. Moreover,

compatibility between the processes and methods employed at the various levels is required to enhance the consistency and traceability. This paper focuses on the transition between the system level and the item level including the processes and methods considered at these levels. This transition is more challenging than the transition between other levels due to the stepover from the systems engineering discipline to the software engineering discipline.

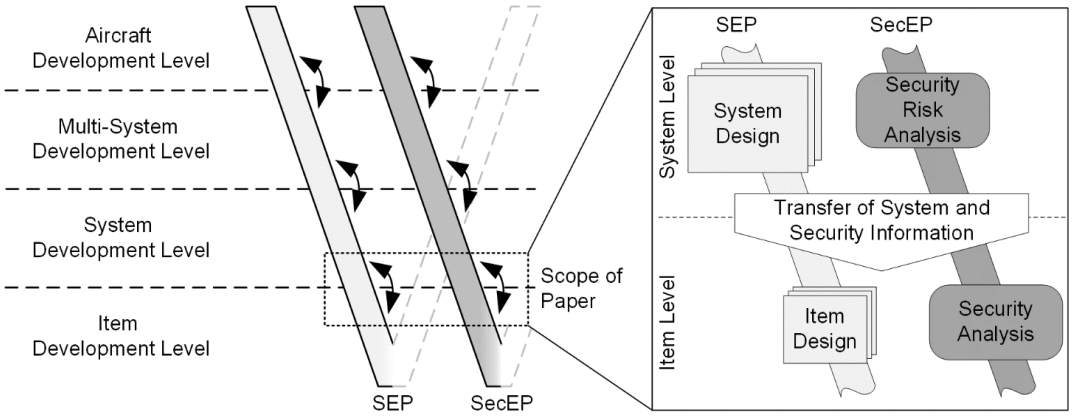

Figure 3: Communication and exchange of information between the various development levels

In both disciplines, domain-specific processes, concepts, languages, and tools are utilized, despite the fact that numerous elements of systems engineering, such as process models, e.g., the V-model, have their origins in software development. Consequently, in order to facilitate the transition between these levels in the context of a holistic security engineering approach, domain-specific aspects of both levels must be harmonized and aligned to be compatible with one another. With regard to the process perspective, in particular, the security risk analysis conducted at the system level and the security analyses conducted at the item level are within the scope. However, as the existing software life cycle processes in accordance with DO-178C do not address the matter of security at all, an analysis of the ED-203A standard and the security assurance objectives defined therein, is required. Based on this analysis, the software life cycle processes can be evaluated against the security assurance objectives and, if necessary, augmented with security-specific activities. The enhanced software life cycle processes are then considered to identify and specify the exchange of information, that is the information that must be provided to the processes at the item level and the information that must be traced bottom-up. The analysis of the security assurance objectives, the defined processes, and the specified information flow serves as a basis for defining process-driven requirements for an approach that facilitates the transition between these levels.

4.1 Analysis of Security Assurance Objectives Associated with the Item Level

The security assurance objectives defined in ED-203A are applicable to the various levels of development. Therefore, the scope of each security assurance objective is defined in the standard, which specifies whether the objective applies to the aircraft, system, or item level. A distinction is made between security-specific assurance objectives and security development objectives. Both assurance objectives are not only intended for aircraft and system level but also require processes and actions at item level. Security-specific assurance objectives are not related to existing (safety) development standards, i.e. these objectives are not dependent on the objectives defined in ARP4754A and DO-178C and, as a consequence are not covered by the associated activities. Security development assurance actions can be covered by development assurance actions when, for example, Intentional Unauthorized Electronic Interaction (IUEI) is considered which is why security development assurances are related to development objectives specified in ARP4754A and DO-178C [10].

The ED 203A standard defines a total of 39 security assurance objectives, 32 of which apply to the item level as well as to other levels. Among the various security-specific assurance objectives, objectives regarding

- identification of vulnerabilities,
- security refutation,
- security deployment, and
- continued security monitoring and management

apply to the item level. Due to the aforementioned focus of this paper, only identification of

vulnerabilities and security refutation objectives are considered. Identification of vulnerabilities encompasses two objectives that address the identification, evaluation and treatment of vulnerabilities in security measures and assets whereas security refutation objectives address refutation analysis and refutation testing in order to identify new vulnerabilities and to challenge the vulnerability evaluation [10]. As defined in ED-203A, refutation analysis is the demonstration of the hypothesis that a design or implementation is free of vulnerabilities. In contrast to verification, refutation activities are not based on requirements, i.e. refutation is not a requirements-based approach. Rather, refutation analysis and testing are based on the attack scenarios that are identified during the security risk analysis and formulate tests] from the attacker's perspective [10]. The identification of vulnerabilities and the objective of security refutation are not related to any DO-178C development assurance activities. Consequently, the extension of current software lifecycle processes is inevitable to satisfy these security assurance objectives.

Unlike the security-specific objectives, the security development assurance objectives are related to development assurance standards such as ARP4754A or DO-178C. These security development assurance objectives may be satisfied by corresponding development assurance objectives with consideration of IUEI [10]. However even security development assurance objectives may require security specific activities which is why conformance with development assurance objectives of the safety process does not exclude additional security-specific evidence to comply with security development assurance objectives [10]. In addition to the security-specific assurance objectives mentioned above, security development assurance objectives, that address

- security requirements,
- design,
- security verification

are considered in this paper. These objectives are closely related to the software development process objectives and the software verification process objectives. The security requirements objectives include security requirements definition and validation. Aside from the development of security requirements according to the related DO-178C objectives, the security requirements, security measure interfaces, and security assumptions have to be defined, as well as bypassing and tampering of the security measures have to be considered in the validation. The design objectives add the definition and validation of a security architecture to the software design objectives specified in DO-178C. A security architecture may be based on the software architecture and defines the architectural elements which implement the security measures to satisfy the security requirements [10]. To satisfy the security verification objectives, in addition to the software verification process objectives, robustness verification elements have to be developed with consideration of abnormal inputs and conditions, i.e. considering IUEI while developing functional and robustness verification elements. Based on the assurance objectives and its associated security-related activities, the software life cycle processes according to [7] are extended in the following section.

4.2 Item Level Security Engineering Process and Software Life Cycle Processes

Today's software life cycle processes as defined by DO-178C do not explicitly address security, i.e., there are no process objectives and associated actions defined that are related to security. Security requirements derived from the security risk analysis at system level are passed to the software life cycle processes as defined in ARP4754A and developed according to the processes and activities specified in DO-178C. In addition to these processes and activities, security-specific actions need to be performed as outlined before to satisfy the security assurance objectives. For this reason, the software life cycle processes are extended by including security specific actions that are closely related to existing software development assurance objectives. Additionally, the security engineering process is ongoing at the item level and encompasses actions that are most likely to be the responsibility of security experts. Both, the established security engineering process and the software life cycle processes are depicted in Figure 4.

Based on the security specific security assurance objectives, the security engineering process at item level involves the analysis of security vulnerabilities and security refutation analysis and testing. The definition of security requirements and its related actions and the design of a security architecture are added to the corresponding software development processes. Both activities are closely related to the existing software requirements and software design process respectively and require only minor adjustments like explicitly tagging security requirements or the characterization of security measures within the software architecture. The security verification activity complements the software verification process and emphasizes the necessity to consider IUEI while developing verification elements and to

develop robustness verification elements considering abnormal inputs and conditions.

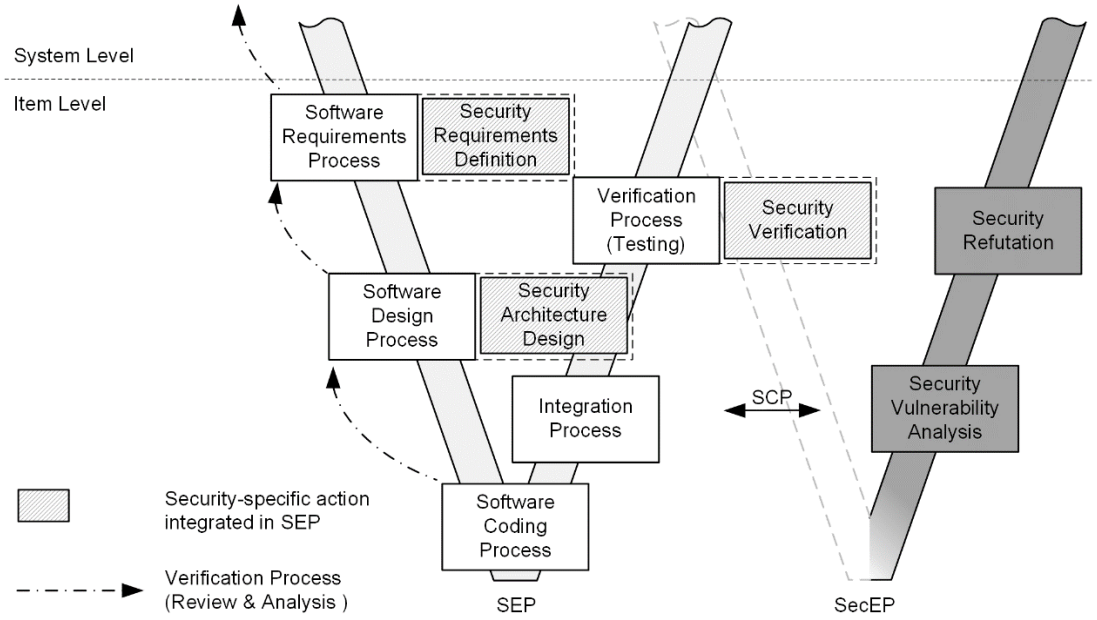

Figure 4: SecEP and software life cycle processes at item level

4.3 Vertical and Transversal exchange of information

The exchange of information between the system level and the item level within the SEP is defined by the ARP4754A and DO178C standard. According to these standards, the information flow from the system process to the software process includes, for example, the requirements that are allocated to the software item, the design assurance levels for the item and the system description Conversely, from the software process to the system process, derived requirements are communicated in the event that they arise at the item level [5]. Consequently, the exchange of information is essentially bidirectional. Compared to the SEP, detailed description of the information flow between the security-related processes at the system and the item level is not provided. The security risk analysis performed at system level aims at identifying and assessing security risks with respect to the system design. Having identified unacceptable security risks, security requirements are defined to introduce security measures and to mitigate the security risk to an acceptable level. Security requirements, thus, represent a significant output of the security risk analysis. To develop secure items, these security requirements need to be communicated to the item level in addition to the functional and other non-functional requirements.

Apart from security requirements, there is more security-related information available at system level. This includes the threat scenarios, which are defined as part of the security risk analysis considering the system architecture. Threat scenarios mainly consists of an origin, vulnerabilities, attack paths, security measures, if present, and threat conditions. These elements of a threat scenario in the context of the respective items can be transferred to the item level enabling the reuse of this information for analysis purposes. Furthermore, the security assurance level (SAL) allocated to the security measures has to be forwarded to the item level as it stipulates the actions required to satisfy security assurance objectives.

The requirements for a comprehensive security engineering approach with a focus on the transition from system to item level can be summarized as follows:

- The approach shall be aligned with the DO-326/ ED-202 Set and fulfill the defined requirements and security assurances objectives within it.
- The approach shall be fundamentally aligned with the ARP4754A and the DO-178C.
- The security requirements and the security assurance level shall be transferred from the system level to the item level.
- The threat scenarios pertaining to the items under development may be transferred from the system level to the item level to facilitate reuse and consistency.
- The information exchange shall be bidirectional.
- The approach shall provide model-based methods to manage the complexity of aircraft systems.

5. Enabling Security Engineering for Airworthiness Through the Use of Modeling

In the context of a holistic security engineering approach, the processes and the exchange of security information described in Section 4 can be effectively implemented by applying appropriate methods. Given the high complexity of aircraft systems, the development is increasingly model-based. For the system development, the semi-formal modeling language SysML is employed. To enhance consistency and to facilitate traceability, security risk analysis is also conducted in a model-based manner. Model-based methods for security risk analysis are described in [31], [32] and make use of the profile extension mechanism of SysML to incorporate concepts used for the security risk analysis into the modeling language. In contrast, UML is accepted and commonly used for software design at the item level. Similarly, approaches to verify and validate security requirements at the software design level are described in the literature [3], [24]. These approaches utilize the extension mechanism of UML. In order to ensure the compatibility of item-level with system-level modeling methods, it is essential to establish an overall harmonized and suitable methodology. For achieving this, it is necessary to first identify and examine methods from the literature that can be used for security analysis at the item level. These methods must be evaluated in relation to their applicability in the aviation context, as described in Section 2. In particular, the use of model-based methods at the item level requires compliance with the DO-331 standard [14], which supplements DO-178C [7] and addresses model-based development and verification. Moreover, when applying formal methods for the verification in combination with corresponding tools, DO-330 [13] and DO-333 [15] must be considered and compliance to these standards must be ensured. Model-based methods, which are generally suitable for this purpose must be adapted to the specific context in order to fulfill the security assurance objectives identified in Section 4. This entails comparing the utilized security concepts with those underlying the security risk analysis at the system level and, when necessary harmonizing them. The harmonization of these concepts is facilitated from the definition of a unified security terminology for the purpose of standardizing the terminology associated with the concepts.

With regard to the system level and the item level processes, the combinational use of various model-based methods within the security engineering process is conceptually illustrated in Figure 5.

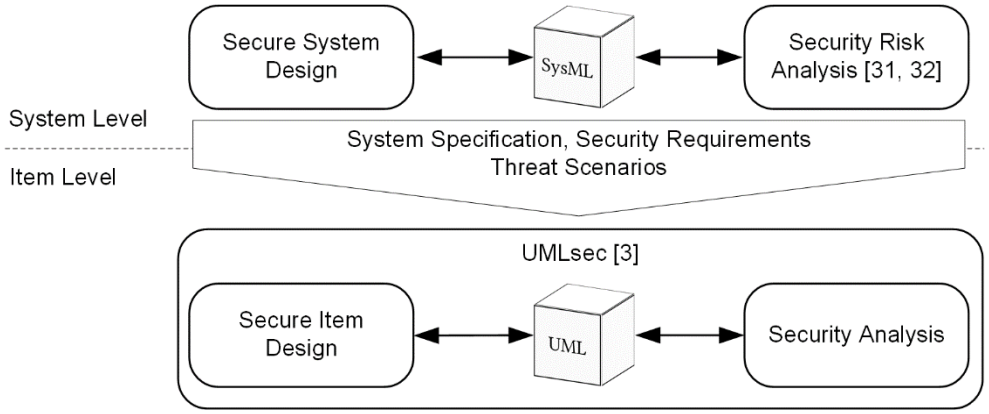

Figure 5: Combination of SysML-based Security Risk Analysis and Secure Software Design by means of UMLsec

In this figure, a SysML-based method for conducting the security risk analysis and integrating it into the system model [31], [32] as well as UMLsec for the design and analysis of secure software systems is used. In this modeling context, the requirements for security engineering for airworthiness can be summarized as follows:

- The approach shall provide a common terminology for the various methods used within the security engineering process at the system and item level.
- The approach shall be built on harmonized security engineering concepts that uses the common terminology.
- The approach shall fundamentally use model-based methods to realize at the very least the software requirements and design processes.
- Given the context of model-based development and verification, the approach must conform to the DO-331 [14] standard.
- Given the context of (model-based) security verification using formal methods and dedicated tools, the approach must conform to the DO-330 and DO-333 standard [13], [15].

This comprehensive model-based security engineering approach addresses the implementation of the processes described in Section 4. The focus is on model-based development using formal or at least semi-formal methods, which require a uniform terminology and concept as well as adherence with the DO-178C standard [7] and the corresponding supplements, *i.e.* DO330, DO331 and DO333 [13]-[15].

6. Conclusion and Outlook

This paper sets out requirements for the transition from the system level to the detailed design and implementation level (item level) with the objective of developing a holistic security engineering approach for airworthiness security. To this end, the regulatory context was introduced and the ED-203A security standard was analyzed. The results of this analysis have led to the formulation of six requirements (cf. Section 4) in the ED-203A context. These include compliance with relevant development and security standards, as well as the exchange of specified information in a bidirectional manner. Moreover, further five requirements (cf. Section 5) have been formulated in a SysML and UML context with regard to the implementation of the identified processes and activities in the context of model-based development.

The elicitation of this whole set of requirements represents the first step in enabling airworthiness security by a holistic security engineering process at various aircraft design levels. Given these requirements, further work will encompass a detailed examination of the supplements to the DO-178C standard, which are particularly pertinent in the context of model-based security design and analysis methods, such as UMLsec and SecureUML.

Acknowledgements

This work was supported by the Federal Ministry for Economic Affairs and Climate Action within the LuFo VI-2 project “Intelligenter und smarter Kabinenbetrieb – Model Based Systems Engineering für eine cyber-sicher zertifizierbare, intelligente und smarte Kabine – FKZ 20D2130K“ based on a decision of the German Bundestag.

Contact Author Email Address

mailto: marvin.blecken@tuhh.de

Copyright Statement

References

[1] David A. Unsettled Topics Concerning Airworthiness Cybersecurity Regulation. SAE EDGE Research Report, 2020.

[2] European Union Aviation Safety Agency. *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes (CS-25)*. Amendment 28, 2023.

[3] Jürjens J. *Secure System Development with UML*. Springer-Verlag, Berlin, Heidelberg, 2005.

[4] Paulitsch M, Reiger R, Strigini L, Bloomfield R. Evidence-Based Security in Aerospace: From Safety to Security and Back Again*. IEEE 23rd International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Dallas, TX, pp. 21-22, 2012.

[5] Society of Automotive Engineers. *Guidelines for Development of Civil Aircraft and Systems*. SAE-ARP4754A, 2010.

[6] Society of Automotive Engineers. *Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment*. SAE-ARP-4761A, 2023.

[7] Radio Technical Commission for Aeronautics. *Software Considerations in Airborne Systems and Equipment Certification*. RTCA-DO-178C, 2012.

[8] Radio Technical Commission for Aeronautics. *Design Assurance Guidance for Airborne Electronic*

Hardware. RTCA-DO-254, 2000.

[9] European Organization for Civil Aviation Equipment. *Airworthiness Security Process Specification*. EUROCAE-ED-202A, 2014.

[10] European Organization for Civil Aviation Equipment*. Airworthiness Security Methods and Considerations*. EUROCAE-ED-203A, 2018

[11] European Organization for Civil Aviation Equipment. *Information Security Guidance for Continuing Airworthiness*. EUROCAE-ED-204A, 2020.

[12] Boehm, B W. Verifying and Validating Software Requirements and Design Specifications. *IEEE Software*, 1, pp. 75-88, 1989.

[13] Radio Technical Commission for Aeronautics. Software Tool Qualification Considerations. RTCA-DO-330, 2011.

[14] Radio Technical Commission for Aeronautics. *Model-Based Development and Verification Supplement to DO-178C and DO-278A*. RTCA-DO-331, 2011.

[15] Radio Technical Commission for Aeronautics. *Formal Methods Supplement to DO-178C and DO-278A*. RTCA-DO-333, 2011.

[16] Airlines Electronic Engineering Committee. *Aircraft Data Network Part 5 Network Domain Characteristics and Interconnection*. ARINC 664P5, 2005.

[17] Airlines Electronic Engineering Committee. *Commercial Aircraft Information Security Concepts of Operation and Process Framework*. Aeronautical Radio, Inc., 2005.

[18] Benz S. *Eine Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil*. 2004.

[19] Hintze H, Wiegraefe B, God R. A Security Engineering Process Approach for the Future Development of Complex Aircraft Cabin Systems. *In Security and Privacy Protection in Information Processing Systems*, L. J. Janczewski, H. B. Wolfe and S. Shenoi, Eds. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, Berlin, Heidelberg, 190–202, 2013.

[20] Mouratidis H, Jürjens J. From Goal-Driven Security Requirements Engineering to Secure Design. *Int. J. Intell. Syst*., 25: 813-840, 2010.

[21] Mouratidis H, Giorgini P. Secure Tropos: A Security-Oriented Extension of the Tropos methodology. *International Journal of Software Engineering and Knowledge Engineering,* 17 (2), pp. 285-309, 2007.

[22] Object Management Group. OMG Systems Modeling Language Specification (SysML). Version 1.6, 2019

[23] Matulevičius R. *Fundamentals of Secure System Modeling*. Springer International Publishing, 2017.

[24] Lodderstedt T, Basin D, Doser J. SecureUML: A UML-Based Modeling Language for Model Driven Security. *In Proceedings of the 5th International Conference on The Unified Modeling Language (UML '02)*. Springer-Verlag, Berlin, Heidelberg, pp. 426–441, 2002.

[25] Mercer E, Slind K, Amundson I, Cofer D, Babar J, Hardin D. Synthesizing Verified Components for Cyber Assured Systems Engineering. 2021 *AMC/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Fukuoka, Japan, pp. 205-2015, 2021.

[26] Society of Automotive Engineers*. Architecture Analysis and Design Language (AADL)*. SAE International, Tech. Rep. AS-5506, 2009.

[27] Roudier Y, Apvrille L. SysML-Sec: A model driven approach for designing safe and secure systems. In *2015 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pp. 655-664.2015

[28] Ameur-Boulifa R, Lugou F, Apvrille L. SysML Model Transformation for Safety and Security Analysis. In *Security and Safety Interplay of Intelligent Software Systems*, Brahim Hamid, Barbara Gallina, Asaf Shabtai, Yuval Elovici and Joaquin Garcia-Alfaro, Eds. Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 35–49, 2019.

[29] Apvrille L, Roudier Y. SysML-Sec: A SysML environment for the design and development of secure embedded systems. In *APCOSEC 2013*, Yokohama, Japan, 2013.

[30] Object Management Group. *Unified Modeling Language (UML)*, Version 2.5.1, 2017.

[31] Hintze H, God R. Using Model-Based Security Engineering in the Development of Complex Aircraft Cabin Systems. *SAE Int. J. Aerospace,* 8, 1, pp. 89-96, 2015.

[32]Hintze H, Speichert J P, God R. The Risk Matrix as an integral part of a SysML-based Security Engineering Approach in the Development of Complex Aircraft Cabin Systems. *In IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, IEEE, 1-9, 2018.