Security and Safety Assurance for Aerospace Embedded Systems

Pierre Bieber, Jean-Paul Blanquart, Gilles Descargues, Michael Dulucq, Yannick Fourastier, Eric Hazane, Mathias Julien, Laurent Léonardon, Gabrielle Sarouille

Security and Safety Assurance for Aerospace Embedded Systems

Pierre Bieber, *Onera*, Jean-Paul Blanquart, *Astrium*, Gilles Descargues, *Thales*, Michael Dulucq, *SERMA*, Yannick Fourastier, *EADS France*, Eric Hazane, *EADS Cassidian Apsys*, Mathias Julien, *Altran*[1], Laurent Léonardon, *Rockwell-Collins France*, Gabrielle Sarouille, *Thales*

Corresponding Author: Pierre Bieber, Onera, 2 avenue Edouard Belin, 31055 Toulouse cedex 04, France *email*: Pierre.Bieber@onera.fr

Keywords: Dependability, Safety, Certification, Security, Standards and Norms, Development and Engineering Process

1 Introduction

1.1 Context

Recent trends in the design of avionics platform increase risks that accidental or intentional misuse of aircraft information may occur. New aircraft platforms have increased the interconnectivity of equipment both within the aircraft and with its environment (aircrafts, satellites, on-ground systems. Such a platform is made of a very wide range of software and hardware items: from highly critical items controlling the aircraft to low criticality items that inform and entertain the passengers through items that help the airline operating and maintaining its fleet. Consequently, the avionics platform could be the target of security issues that could have an impact on the aircraft safety.

Airworthiness has to be ensured even in the presence of aircraft information misuse. In the past ten years, aircraft industry, airworthiness certification authorities and research organizations have been working to deal with this important matter. New functions were designed to protect avionics platforms, regulations addressing security were issued and joint working groups were established to build applicable standards. In particular, EUROCAE Working Group 72 has published in October 2010 the ED202 document [1] that defines a security process for airworthiness.

In that context, partners of the SEISES project have investigated, from October 2008 to December 2011, assurance aspects of the development of secure and safe embedded aerospace systems. This paper details two outcomes of the project: a joint framework that groups and organizes security and safety assurance activities and the lessons learnt by applying this framework on three demonstrators.

1.2 Overview of the Paper

The paper starts with the list of basic principles that guided the development of the SEISES security and safety assurance framework. Then we present the SEISES structure and we provide some examples of assurance objectives and related assurance activities. We detail the convergence between safety and security assurance activities that we have identified. Finally, we introduce the three demonstrators and we summarize the main lessons learnt from these experimentations. We conclude the paper by summarizing the results of the SEISES project, by comparing these results with other approaches dealing with joint safety and security assurance and by listing promising directions for further research.

[1] On Behalf of AIRBUS

2 SEISES Framework Design Principles

2.1 Design Principles

The basic principles that guided the design of SEISES framework are:

- *Processes should deal with "Security for Safety"***:** SEISES processes focus on security objectives whose loss has a safety impact because these objectives have be considered in the scope of an aircraft certification. As the SEISES framework is applicable to the development of safety critical systems, it is assumed that the development follows the classical aeronautical process as described in ARP4754 [2] and DO178 [3].

- *Processes should focus on development assurance***:** SEISES processes deal with the development stages of systems. Later stages of the life of the systems such as delivery or operation are not considered.

- *Processes should be described by a set of assurance objectives***:** The organization and wording of SEISES assurance objectives is consistent with DO178B and ARP4754. So engineers familiar with critical aircraft system development should be able to quickly understand how SEISES assurance objectives relate with the objectives they are used to deal with.

- *Assurance objectives should be implemented using assurance activities described in existing standards***:** Activities that implement the assurance objectives are extracted from standards that were considered relevant by industrial partners.

- *Convergence between safety and security assurance activities should be described***:** SEISES processes establish under which conditions security and safety assurance activities could be shared, this should help to decrease the cost of applying security and safety assurance methods. SEISES processes identify the main interactions between objectives and activities as, even if activities cannot be shared, this should help to improve the consistency between the safety and security views of the system under development.

2.2 Relevant Standards

Relevant standards for the SEISES project are the one that describe processes followed in the aeronautical[2] industry for the development of critical hardware and software systems: ARP4754, DO178B, DO254 [4] for safety aspects, ED202, ISO 27005 [5] and CC v3.1 [6] for security aspects.

These processes can be organized into 8 blocks that group processes dealing with the same aspect (either safety or security), that guide the same kind of activities (either assessment or development) and that is applicable at the same level (either system or item).

[2] Processes applicable in the space industry are also relevant. In this document, we suppose that it will be relatively easy to adapt to the space domain the principles selected for the aeronautical systems.

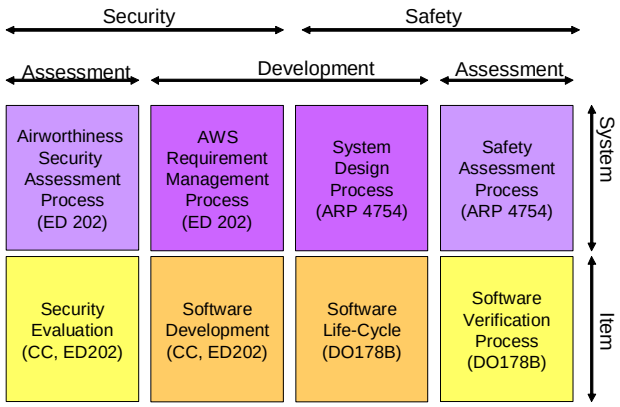

Figure 1: Relevant Existing Processes for SEISES

Standards are made of several processes, some of them are almost exclusively related with assessment activities and others are related with design activities. For instance, ARP4754 describes in parallel the System Design Process and the Safety Assessment Process. Similarly, DO178B describes design processes such as the « Software Development » and assessment processes such as the « Software Verification Process ».

In the aeronautical domain, different standards are applied for system development (ARP4754) and for item development (DO178B, DO254). The distinction between item and system levels does not appear in security documents as ISO27005 or CC. In the aeronautical domain, the distinction is important because activities at system level are usually performed by the airframer whereas activities at item level are performed by suppliers in charge of implementing the equipment.

ARP4754 and ED202 also cover aircraft level activities that are performed before systems are developed. In the context of the SEISES project, we have decided not to consider assurance activities performed at aircraft level because the demonstrators studied during SEISES are either systems or items.

3 SEISES Process Framework

3.1 Organization of SEISES Framework

The three groups of processes containing SEISES assurance objectives are:

- *Risk Assessment* group that includes two processes: Risk and vulnerability analysis and Security Level Determination. These processes contain assurance objectives that aim at establishing the requirements and the efficiency level of the security functions to be developed.

- *Assurance-Effectiveness* group that includes the Security Properties process. This process contains objectives that aim at showing that the functions of the system are efficient to counter the threats that were identified.

- *Assurance-correctness* group that includes Planning, Requirements, Design and Implementation, Validation, Verification, Configuration management, Certification liaison. These processes contain objectives that aim at showing that the functions that counter the threats were correctly developed.

For each process a table was built. It lists assurance objectives to be achieved, and it indicates the assurance activities belonging to the relevant standards that can contribute to the achievement of the objective.

3.2 Risk Assessment

The *Risk and vulnerability analysis* process contains 5 objectives that could be applied at system and item levels.

	SEISES Assurance Objectives	ISO 27005 Activities	CC Activities	ARP4754 Activities
2	*Threat conditions have been defined*	RiskIdentification_4		SafetyAssessment_1

Table 1: Extract from Risk and Vulnerability Assessment Objectives and Activities

Objective 2 aims at defining Threat Condition that is, according to ED202, "*An adverse condition of the aircraft which can result from an information security threat*". This can be achieved by using activities that are extracted from an interpretation of ISO 27005 to the aerospace domain. RiskIdentification_4 groups the following assurance activities: "*The threats and their sources should be identified. The existing controls or security measures should be identified. For each selected assets, the vulnerabilities that can be exploited by a threat to cause harm to assets should be identified*".

A safety assessment activity extracted from ARP4754 such as Functional Hazard Assessment (SafetyAssessment_1[3]) is also relevant to comply with the objective. It contributes indirectly to the objective as it provides information such as the list of failure conditions that is useful to define Threat Conditions. The contribution of CC activities to achieve the Risk Assessment objectives is very limited because the CC activities are applied after Risk Assessment has produced at set of security objectives. DO178B was not considered as it does not contain risk analysis activities.

An important objective of Risk Assessment is to allocate a Security Level ranging from E (lower) to A (higher) with the system functions. This level is used to select applicable objectives. The allocation rules for the Security Level and a comparison between Security and Safety Levels are described in a companion paper [7].

3.3 Assurance-Effectiveness

The following table describes one out of the 5 objectives that belong to the *Assurance-Effectiveness* process contains.

	Objectives	CC Activities	ARP4754 Activities	DO178B Activities
1	*Security Architecture is described and justified*	ADV_ARC	Development_5	Development _3
4	*Penetration tests are written and performed*	AVA_VAN	Implementation_ verification_2,3,4 ,5,6	Testing_1,2

Table 2: Extract from Assurance-effectiveness Objectives and Activities

[3] SafetyAssessment_1 refers to objective 1 in the Safety Assessment table of ARP4754. We use the same approach to name DO178 objectives.

Objective 1 is mainly achieved using activities in the class ADV-ARC of the Common Criteria that deals with the analysis of the security architecture. These activities should be coordinated with ARP4754 activity Development_5 "*System architecture is defined*", and DO178 activity Development_3 "*Software architecture is developed*". Objective 4 can be achieved using activities in the class AVA_VAN of the Common Criteria that is related with Vulnerability assessment. These activities should be coordinated with the implementation verification activities belonging to the ARP4754A and Testing activities in DO178.

These objectives are applicable at all Security Levels. But, for objective 4, the higher the Security Level is, the more powerful the attacker should be considered when performing penetration tests. Similarly for objective 1, the justification of the security architecture will be more detailed at higher Security Levels than at lower Security levels.

3.4 Assurance-Correctness

The following table shows some objectives from the Requirements (one out of 4 objectives), Design and Implementation (one out of 3 objectives), Validation (one out of 4 objectives) and Verification Processes (one out of 4 objectives).

	SEISES Assurance Objectives	**Applicability**				**CC Activities**	**DO178C Activities**
	Requirements	**A**	**B**	**C**	**D**		
1	*Security Functional Requirements (SFR) are developed and tagged*	*R*	*R*	*R*	*R*	ADV_FSP	Development_1,2,4
	Design and Implementation	**A**	**B**	**C**	**D**		
1	*Design requirements are developed from functional specification*	*R*	*R*	*R*	*A*	ADV_TDS, ADV_IMP, ADV_INT	Development_1,2,4,5,6
	Validation	**A**	**B**	**C**	**D**		
4	*Code is validated*	*R*	*R*	*R*	*A*	ADV_IMP	Code_Integration_Verification_1,2, 3,4,5,6,7
	Verification	**A**	**B**	**C**	**D**		
1	*Functional test procedures are written*	*R**	*R*	*R*	*R*	ATE_FUN	Testing_1

Table 3: Extract from Assurance-Correctness Objectives and Activities

Objectives in Assurance-Correctness processes are achieved using similar CC and DO178 activities. CC activities mainly belong to the ADV class that deals with the development (ADV_FSP for Specification, ADV_TDS for Design, ADV_IMP for implementation) and to the ATE class that deals with functional tests. For instance, objective 1 of the Requirement Process can be achieved either using DO178B objective "*High-level requirements are developed*" or using activities in the class ADV_FSP of the Common Criteria as ADC_FSP.FSP.1.1D "*The developer shall provide a functional specification*".

The table also defines the applicability of objectives for a given security level, this definition is consistent with what DO178 requires for a given DAL.

4 Convergence between Safety and Security Assurance

4.1 Global Safety and Security Process

To study the convergence of Safety and Security Assurance activities we have defined a global process that merges safety and security assurance. This global process is described in figure 3. The figure is organized in four columns, on the left side we find two columns related with Security activities (one for the development activities and the other for assessment activities) and in the right side we find two columns for Safety activities. The figure is organized in 8 rows that group activities developing the same kind of description of the system or that assess the correctness of the developed description.

The ordering of activities from top to bottom in each column provides an indication of the sequencing of activities. For instance, the development of a system is decomposed in 8 steps : aircraft functions are allocated to the system, a system architecture is developed, requirement for items are derived, these requirements are transformed into high-level requirements for the software, that help to design a software architecture and low-level requirements for the software, a source code is produced to implement the low-level requirements, an executable code can be produced, this code is finally integrated in the system architecture.

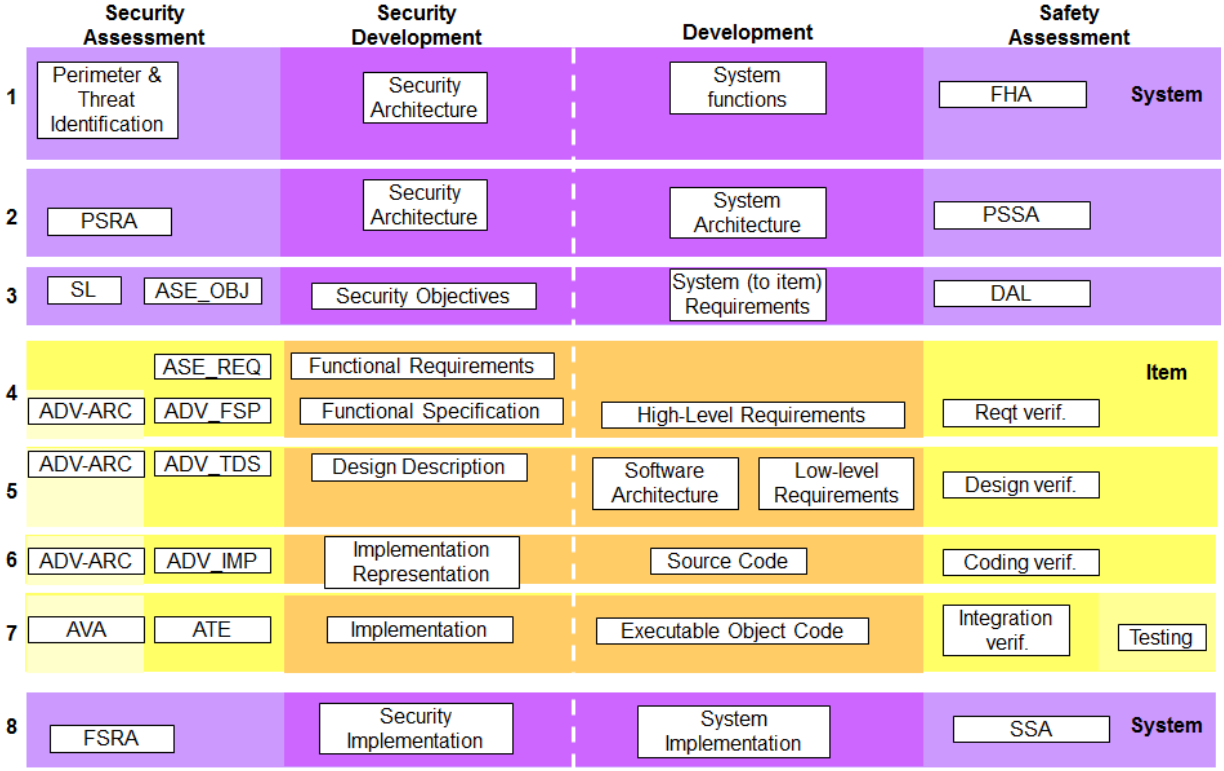

Figure 2: Global Safety and Security Process

Some development activities belonging to the same row could be performed only once. As security development is a part of the development activities used in the safety process, a lot of security development activities are already performed by the safety process. For instance, in row 7 (resp. 8), developing the "*source code*" (resp. "*executable object code*") should be equivalent to developing the "*implementation representation*" (resp. "*implementation*").

Currently, some activities might be slightly different in the two development processes. For example, the definition of security objectives, requirements and specifications uses terms as “Target of Evaluation” (TOE)[4], “TOE Security Functional Requirements”, “Security Enforcing Functions” that are currently not shared with safety development. But one could hope that when system designers will become more familiar with security considerations, these terms will become a part of the classical development process. So it is likely that in the future the two middle columns in the previous figure will be merged into one column that would describe development activities that take into account both security and safety considerations.

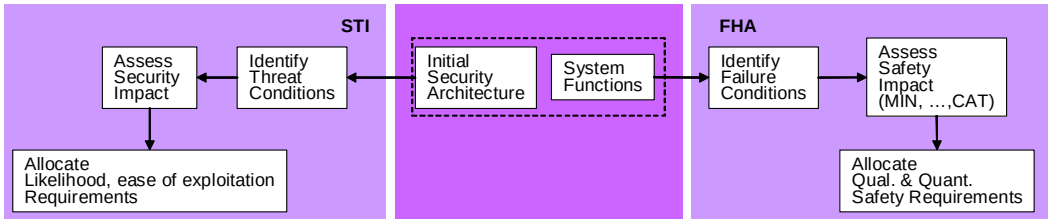

Figure 3: Links between Security and Safety Risk Assessment activities

Assessment activities belonging to the same row should interact. In order to be consistent, these activities should at least share the description of the system being developed at this stage. Let’s consider the case of assessment activities in row 1: Functional Hazard Assessment and Threat Identification. First of all, a common preliminary description of the system under development should be shared by these activities. This description should include the system function breakdown that is used to identify Failure Conditions in the safety process. The functional breakdown should also guide the selection of assets in the security process. Furthermore, activity RiskIdentification.4 “*The safety impact of loss of security should be assessed for each asset*” needs as input the Failure Condition classification that is the output of the FHA activity. Conversely, it could be interesting to add a step in the FHA process where the safety analyst would review the Threat Conditions and check whether the list of Failure Conditions should be extended in order to take into account security risks.

4.2 Similarities and Differences between Safety and Security Assurance

A number of similarities and differences between safety and security assurance appeared when the SEISES framework was built. With respect to assurance-correctness objectives, there are strong similarities between DO178 activities in the Development, Requirement, Design, Coding, Verification processes and CC activities in the ADV and ATE class. As it was already mentioned, CC activities use specific terms as “Security Supporting Functions”, “Modules” and “Sub-Systems” that are not used in DO178. Planning activities in DO178 can be related with ALC_LCD class activities. Configuration activities in DO178 can be related with ALC_CMC and ALC_CMS classes. Some CC assurance-correctness activities in the AGD class on user guidance documents or ALC_DVS on the security of the development environment, ALC_DEL on the delivery of the TOE cannot be directly related with activities in DO178.

There are no obvious counterparts to assurance-effectiveness objectives in the safety assurance processes. Nevertheless it can be argued that any activity that aims at assessing the design or verifying the implementation of the system security architecture could contribute to assurance effectiveness.

The respective role of the developer and of the certification body is different in airworthiness certification and in security certification. In airworthiness certification, the developer and the certification body first negotiate an acceptable development process. The result of this negotiation is

[4] The TOE is the IT object submitted to a thorough expertise because of an identified security issue.

formalized by a Design Organization Approval that delegates from the Authority to the Industry the evaluation activities. Then the developer is in charge of both the development and the evaluation of the developed item. Yet, it could be the case that critical assurance activities related with security testing are performed by an entity that is independent from the development team. The certification body may perform some audit of the development but it does perform all the evaluation tasks. Whereas in the security evaluation schemes underlying the Common Criteria, the certification body (or one of its proxies) is involved in the evaluation of the developed item. To be able to use CC activities in an aircraft development, most of the activities that describe evaluator actions have to be performed by the developer. Some activities requiring specialized expertise as vulnerability assessment activities in the class AVA_VAN could still be performed by an evaluation entity.

5 Three Demonstrators

5.1 Demonstrator Descriptions

Three demonstrators were developed by the industrial partners from the SEISES consortium. The demonstrators aim at experimenting with the security and safety assurance framework. The selection of demonstrators covers several aspects of the SEISES framework (item/system level assurance, various criticality and security levels).

Demonstrator D1 is an existing communicating item with medium criticality (it contains DAL D and DAL C software) and medium security needs (similar to EAL3). This demonstrator aims at experimenting with item level activities. This demonstrator was developed by Rockwell-Collins France. As the demonstrator is an already existing equipment, the experimentations consisted in comparing activities that were actually performed during the development of the demonstrator with objectives that are mandated by the SEISES framework.

Demonstrator D2 is an existing part of the Command and Display System that is in charge of communicating video signals to be shown on the crew displays. This part of the system is made of several occurrences of high criticality (DAL A) and medium security (similar to EAL4) equipment. This demonstrator aims at experimenting with item level activities. This Demonstrator was performed by Thales. Again the experimentations consisted in comparing activities that were actually performed during the development of the demonstrator with SEISES objectives and activities.

The last demonstrator, D3, is related with architectural studies for future generation Cockpit system. This involves items with a moderate criticality and high security needs. This demonstrator aims at analyzing assurance objectives and activities related with Risk Assessment. This demonstrator was performed by Airbus. As this system is not already implemented, the experimentations focused on the feasibility of the proposed assurance activities.

5.2 Lessons Learnt

Demonstrator D1 used a preliminary version of the SEISES framework that listed the assurance objectives but did not contain the activities. SEISES assurance objectives were validated by showing that they were consistent with what was performed during the development of D1. An important lesson learnt from D1 was that developers needed guidance in order to select assurance objectives that should be applied. A second version of the framework was then developed. It uses the Security Level to define what assurance-effectiveness and assurance-correctness activities should be performed.

Demonstrator D2 used the second version of the framework. D2 validated SEISES assurance activities by comparing them with the activities performed for the development of D2. Commonality between CC and DO178 activities was investigated. One lesson learnt from D2 was that the wording of some CC activities needed to be adapted in order to enhance their consistency with the aeronautics

practices. Another lesson learnt was that guidance about the scheduling of activities was missing. To correct this limitation Thales proposed to map SEISES assurance objectives onto ED202 processes.

Demonstrator D3 also used the second version of the framework. D3 first performed a risk analysis for a future system. It also studied assurance activities that deal with system architecture design. For these two activities D3 investigated the interactions between the safety and security processes. D3 confirmed that Safety and Security assurance activities should share the same system design description. Furthermore D3 showed that safety barriers could play an important role when designing architecture. Consequently, one lesson learnt from D3 is that functions that play a barrier role should be shared between safety and security assessment.

6 Concluding Remarks

Other groups have studied frameworks combining assurance activities for Safety and Security: EUROCAE WG72 that produced ED202 [1], SQUALE European Project [8], SAFSEC Project [9], University of Idaho [10]. The SEISES framework follows the same principles as ED202 because both approaches propose to introduce security considerations into a development process that is focused on safety. The approach proposed by SQUALE could be seen as the reverse option: introducing dependability concerns into a development process focusing on security. In particular, SQUALE proposed to extend the notion of security target into a dependability target.

The SEISES framework provides an organized collection of safety and security assurance objectives that is consistent with the current practice in safety critical system development. Thanks to the demonstrator experiments using the SEISES framework we are confident that it provides a sound foundation for an applicable development process.

Further work should be performed before the SEISES framework is applied in Industry. The framework should be extended in order to take into account important security considerations as secure delivery, administration guidance. Methods and tools supporting some of the assurance activities such as, for instance, threat scenario identification should be developed. Another topic that needs to be addressed is how to deal with Commercial Off The Shelf (COTS) components that were not developed according to the SEISES framework.

7 References

[1] EUROCAE WG72, "*ED202 - Airworthiness Security Process Specification*", October 2010

[2] SAE AEROSPACE, AEROSPACE recommended practice Development of civil aircraft and systems ARP4754a Rev J1

[3] RTCA, Software Considerations in Airborne Systems and Equipment Certification DO178B December, 1st, 1992

[4] RTCA, Design Assurance Guidance for Airborne Electronic Hardware DO254, April 2000

[5] ISO/IEC 27005:2011 Technologies de l'information - Techniques de sécurité - Gestion des risques liés à la sécurité de l'information, 2011

[6] Common Criteria for Information Technology Security Evaluation - Part 1: Introduction and general model, Version 3.1 Revision 3 Final CCMB-2009-07-001 July 2009

[7] SEISES Consortium, "*Safety Levels and Security Levels*", to appear in the proceedings of ERTS^2 2012.

[8] SQUALE Consortium, "SQUALE Dependability Assessment Criteria", 1999. http://www.laas.fr/TSF/cabernet/squale/SQUALE4.pdf

[9] Samantha Lautieri, David Cooper, David Jackson "SafSec: Commonalities Between safety and Security Assurance" in the proceedings of the Safety Critical Systems Symposium, 2005

[10] Carol Taylor, Jim Alves-Foss, Bob Rinker "Merging Safety and Assurance: the Process of Dual Certification of Software" in the proceedings of Software Technology Conference, 2002.