Evidence-Based Security in Aerospace

From Safety to Security and Back Again

Michael Paulitsch Rupert Reiger
EADS Innovation Works
Munich, Germany

Lorenzo Strigini[1] Robin Bloomfield[1,2]
[1]Centre for Software Reliability, City University London
[1,2]Adelard LLP London, United Kingdom

***Abstract* — Security concerns of safety-critical systems increase due to interconnections of systems. This paper tries to outline future security requirements in avionics and issues in assessing the reliability of software from the safety and security perspective. Quantitative work on software reliability has focused on requirements-to-code translation; software security has focused more on requirements correctness. Future work must take advantage of results from both the security and safety areas.**

Keywords*—safety; security; avionics; software assurance*

I. INTRODUCTION

Security is becoming a major concern of aircraft manufacturers as airplane connectivity is increasing to new levels. Connectivity off-board comprises passenger connectivity, automatic content refreshes (maps, entertainment systems), and maintenance activities in addition to traditional control tasks (e.g. for air traffic management). Security also concerns onboard systems and their separation and information flow. ARINC811 describes aircraft operations and maintenance considering security aspects [6]. Figure 1 depicts aircraft network security domains, major aircraft system, and access properties (closed, private, and public) and users of domains.

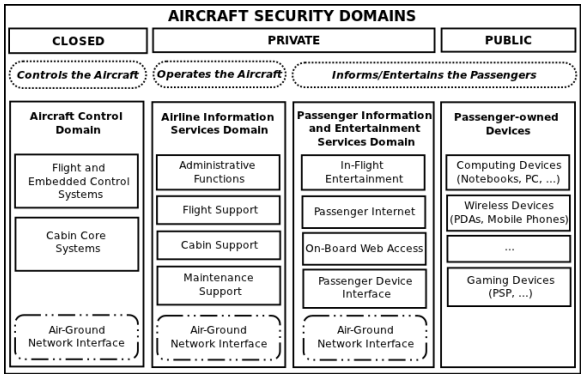

Figure 1. Onboard aircraft network domains (ARINC-811 [6])

Future aircraft architectures aim to increase connectivity between domains and ground services. With this, security concerns increase as described exemplary in [1][2] and system development requires architectural elements ensuring security, such as secure information flow using gateways [3]. A major goal of security is to ensure the safety of the passengers and crew, with confidentiality taking lower priority. Hence, aircraft IT security is best described as "security for safety" [7].

Secure architectures are only one element to achieve secure operations. In addition, aircraft operators need to use security processes, evaluating and monitoring risks and vulnerabilities and considering (new) threats to critical assets [1] [6]. The implementation of the architecture also has primary importance in determining security. E.g. "*the effectiveness of a security countermeasure is its ability to reduce the occurrence of successful attack and is the combined result of the strength of mechanism and the implementation assurance, as classified by the security level of the system or item.*" (p.22 [5]) Assurance of the implementation is thus as important to security as the architecture itself.

The aerospace sector has developed strong assurance processes to ensure safe implementation, such as DO-178/ED-12 [8] for software and DO-254/ED-80 [9] for hardware. A major question now is to what extent these processes are also effective in achieving security. We aim at collecting (ideally quantitative) evidence from existing safety-related and security-related processes and reason about if and how indicators of effectiveness of safety-related processes and methods are useful predictors for their effects on security in aircrafts. For example, whether despite the benign threat models often applied in the safety domain, the software engineering that removes safety-critical implementation defects also removes security vulnerabilities.

II. SAFETY AND EMERGING SECURITY STANDARDS IN AEROSPACE

A. *Safety*

Figure 2 shows the safety-related development assurance process in aerospace. ARP 4754 [11] gives guidance for system level development to fulfill legal requirements of safety (like FAR 25.1309 for transport category aircraft). It defines steps for adequate refinement and implementation of requirements. ARP 4761 [12] describes adequate safety assessment during this refinement, via techniques like FMEA (Failure Mode and Effects Analysis), FTA (Fault Tree Analysis), Functional Hazard Assessment (FHA), (Preliminary) System Safety Assessment ((P)SSA), and Common Cause Analysis (CCA). Besides ground and flight tests, analysis including engineering analysis, stress analysis, system modelling and similarity modelling contribute to ensure safety.

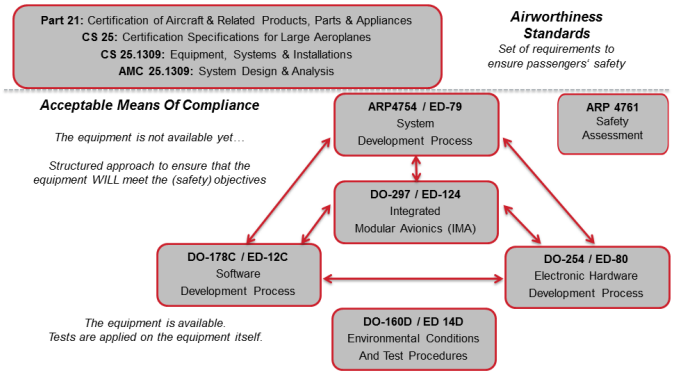

Figure 2. Overview of Safety-Related Development Assurance in Aerospace

For software, DO-178 describes the process to transform requirements to code, assuming the requirements (input) as correct. Techniques used to ensure correctness are independent reviews, traceability, robustness tests, etc. It should be noted, however, that the code is only as correct as the requirements. Esposito and others provide an overview of major safety standards [10].

B. *Security*

Security processes for IT are emerging in aerospace. ED202 [5] provides guidance to developers and certification authorities for aviation systems that can be adversely affected by human interactions and can eventually impact the aircraft's safety properties. For this purpose, the standard establishes a relationship between the impact of a threat scenario, the threat scenario's likelihood, and the final risk-acceptance value for the aircraft system. If the likelihood of a threat scenario is too high to be acceptable, either the threat scenario's likelihood or its impact must be decreased to meet safety requirements.

The Common Criteria [13] also apply in aerospace, e.g. by using separation kernel protection profiles for embedded operating systems in aerospace such as the work of the Open Group on compositional approaches to security and MILS Common criteria focus on ensuring correctness of requirements and system architectures. Heitmeyer and others describe an example applicable to aerospace in [17].

III. DISCUSSION

While there is research on the effectiveness of software engineering techniques, quantitative results based on field data are rare. Software processes in aerospace concentrate on correctly translating requirements to code and less on the verifying the correctness of the code itself, although there is considerable work on abstract interpretation and showing the absence of certain forms of run-time exceptions.

Safety is a system property, but available work on process evaluation is focused mainly on the software process. Among reports on field data, German describes a major technique, software static code analysis, and its effectiveness [15]. An anomaly discovery rate of 1 in 1000 lines is not uncommon (though does not mean that this has a direct safety effect), however we are more concerned with judging possible defects left in the code, not those removed. Data on effectiveness of static analysis and other software engineering techniques in industrial application are sparse. In addition, there is evidence that the discovery rate of vulnerabilities is closely related to the effort spent on seeking them and in some common products there is an effectively infinite pool of vulnerabilities. Some techniques are available for estimation of residual errors in safety-critical software, such as Bloomfield and Guerra present in [16]. However the extension of this work to link process to achieved security-related software reliability is a major challenge given the dynamics of the threat environment.

Carter summarizes a workshop on safety-critical versus security-critical software [14] describing that techniques for determining safety and security requirements are essentially the same. The security domain would benefit from using software development techniques from the safety domain but cost and time-scale implications may be a barrier.

In software security, the focus is often split between adapting to addressing recent vulnerability patches and overall correctness of the software requirements. However, a major need is to assess the translation of software requirements to code.

The authors are members of the SeSaMo collaborative project (http://sesamo-project.eu), addressing the combined assessment of safety and security in embedded systems. An important component of the problem in aerospace is to collect evidence on the effectiveness of safety-driven processes, and security work based on Common Criteria evaluations, ideally including composite certification aspects [4] and to develop models to allow us to understand and judge deployed systems. This is enormously ambitious but given the threat reality something that is urgently needed. Any modeling and associated evidence must take into account both the challenges of aleatory and epistemic uncertainties.

A major question is to assess (ideally quantitatively) how effectively existing safety-oriented processes are performing from the viewpoint of security and how effective (new) security informed processes might be. Given the relatively strong performance of safety-oriented software processes in requirements-to-code translation, and in requirements checking (modeling of software requirements to ensure correctness), the question remains whether safety-related processes for translation also are effective in preserving security properties.

ACKNOWLEDGMENT

This work is part of the SeSaMo project, supported by the Artemis JU, the German Ministry of Education and Research (funding ID 01IS12003), and the United Kingdom Technology Strategy Board (ID 600051 and 600052). The responsibility of the content lies with the authors. Special thanks to Peter Ladkin for his input.

REFERENCES

[1] G. Ladstaetter, N. Reichert, and T. Obert, "Securing IT in the Sky," SAE electronics and connectivity, Society of Autom. Engineers, Sept. 5th, 2012.

[2] A. Dessiatnikoff, Y. Deswarte, É. Alata, and V. Nicomette, „Potential Attacks on Onboard Aerospace Systems. IEEE Security & Privacy Magazine, pp. 71-74. July/August 2012.

[3] K. Müller, M. Paulitsch, S. Tverdyshev, H. Blasum, "MILS-Related Information Flow Control in the Avionic Domain: A View on Security-Enhancing Software Architectures," Workshop on Open Resilient human-aware Cyberphysical Systems (WORCS), IEEE, 2012.

[4] K. Müller, M. Paulitsch, R. Schwarz, S. Tverdyshev, H. Blasum, "MILS-based Information Flow Control in the Avionic Domain: A Case Study on Compositional Architecture and Verification." In Proc. of the Digital Avionics Systems Conference. AIAA/IEEE. Oct. 2012.

[5] The European Organisation for Civil Aviation Equipment, "ED-202 / DO-326 – Airworthiness Security Process Specification," 2010.

[6] Airlines Electronic Engineering Committee, "ARINC 811: Commercial Aircraft Information Security Concepts of Operation and Process Framework," Aeronautical Radio, Inc. 2005.

[7] B. Triquet, "Mixed Criticality in Avionics," Position Paper, European Commission Workshop on Mixed-Criticality Systems, Brussels, Belgium. Feb. 3, 2012.

[8] RTCA, "DO-178C/ED-12C - Software considerations in airborne systems and equipment certification," RTCA, Inc, Dec. 2011.

[9] RTCA, "DO-254/ED-80 - Design assurance guidance for airborne electronic hardware," RTCA, Inc, 19th April 2000.

[10] C. Esposito, D. Cotroneo, N. Silva, "Investigation on Safety-Related Standards for Critical Systems," *Software Certification (WoSoCER), 2011 First Int. Workshop on* , vol., no., pp.49-54, Nov. 29 2011-Dec. 2, 2011.

[11] Society of Automotive Engineers (SAE), "ARP (Aerospace Recommended Practice) 4754: Certification Considerations for Highly Integrated or Complex Aircraft Aystems,", Revision A, Dec. 2010.

[12] Society of Automotive Engineers (SAE), "ARP (Aerospace Recommended Practice) 4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment", 1996.

[13] Common Criteria Sponsoring Organizations, July 2009, Common Methodology for Information Technology Security Evaluation, Version 3.1, rev. 3, http://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R3.pdf.

[14] A. Carter, "Safety-critical versus security-critical software," System Safety 2010, 5th IET Int. Conf.on , vol., no., pp.1-6, 18-20 Oct. 2010.

[15] A. German. "Software Static Code Analysis - Lessons Learned," Crosstalk. Vol. 16 (11), Nov. 2003.

[16] R.E. Bloomfield and S. Guerra. "Process Modelling to Support Dependability Arguments". In Proc. of the 2002 Int. Conf. on Dependable Systems and Networks. Washington, DC, USA, 2002.

[17] C. Heitmeyer., M. Archer, E. Leonard, J. McLean, 2008, Applying Formal Methods to a Certifiably Secure Software System, IEEE Trans. Softw. Eng., vol. 34, IEEE Press, pp. 82–98.