Security Assurance for IT Infrastructure Supporting Airplane Production, Maintenance, and Operation

Scott Lintelman*, Richard Robinson, Mingyan Li, Boeing; David von Oheimb, Siemens Corporate Technology; Radha Poovendran, Krishna Sampigethaya, University of Washington

Abstract

The present paper seeks to motivate interest in a Boeing-led workshop session on the security evaluation of IT infrastructure supporting aircraft, with focus on issues that arise when flight-critical software and related data are distributed among suppliers, manufacturers, airlines, and maintenance organizations via open information networks. We would like to share our pioneering experiences in developing security requirements for an airplane asset distribution system, and to receive feedback from other parties involved with aircraft IT infrastructure. We propose to collaboratively develop guidelines for system certification requirements and to identify directions for future research.

Keywords

Airplane software, information technology, infrastructure, software engineering, safety, security, certification

1. Introduction

The utilization of computer networks for electronically storing and distributing loadable software airplane parts (referred herein as *parts*) [1] and airplane data, such as health management data [2], enables time-efficient and cost-effective airplane production and maintenance processes, compared to legacy part and data distribution based on the physical transfer of storage media [3]. As aircrafts become more connected with their online environment, opportunities for security attacks are opened up. Moreover, since COTS hardware and software components are becoming more widely used, the potential for re-engineering and sabotaging aircraft IT components rises. In effect, the industry's large investment into the safety and reliability of airplane software is at risk.

In particular, electronic storage and distribution mechanisms may have vulnerabilities that afford exploitation by attackers to lower airplane safety or to compromise systems that, while they imply no safety impacts, nevertheless may affect passenger comfort and confidence or airline business processes. Any such vulnerability has the potential to negatively impact the businesses involved with airplane production or maintenance, for example, by creating unwarranted delays. These considerations motivate a requirement to identify and address potential threats to safety and business that arise specifically from electronically storing and distributing airplane parts and data.

1.1 Airplane Assets Distribution System (AADS)

We refer to an overall system used to store and distribute airplane parts and data (together referred as *assets*) as an Airplane Assets Distribution System (AADS). Fig. 1 presents a generic AADS system model. The entities include: *(i) Airplane* that receives software parts and generates health data, *(ii) Organizations* that include supplier of airplane assets, manufacturer of airplane, owner of airplane, and service provider maintaining the airplane. Fig. 1 also shows the flow of assets among these business entities.

An AADS stores assets and forwards them from a source to a destination. The distribution process involves the request and release of stored assets at the source entity, the delivery of assets, and the reception and inspection of assets before storage at the destination entity. We have assumed that a supplier is accountable to provide safety-assured and correctly functioning assets, and that the target hardware (e.g. Line Replaceable Units (LRUs) for processing the assets onboard the airplane) are trusted. As shown in Fig. 2, an *adversary* is any malicious entity external or internal to AADS, seeking to lower the safety of airplane or impede the business of the remaining entities by manipulating the assets at intermediate entities.

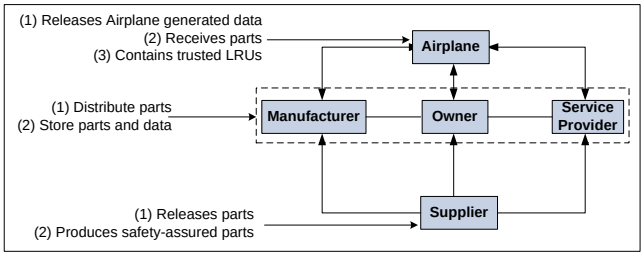

Fig. 1. Airplane Asset Distribution System (AADS).

2. Security Threats to AADS

Certain assets that are loaded to an aircraft, e.g. flight control computer software, have safety implications, as evident from the standards and advice mandated for airplane loadable software development assurance in [4]. The integrity of such safety-critical assets must be protected at all times. Therefore, with the use of public networks for storing and distributing these assets in AADS, the following threats must be addressed.

2.1 Threats to Airplane Safety

In an attempt to lower safety of the airplane, an adversary may try to corrupt safety-critical assets during their distribution and storage in the AADS. For example, an adversary

* Corresponding author: Scott A. Lintelman; Email: scott.a.lintelman@boeing.com, Tel: 425-373-2611; Mailing address: 7L-49, 2760, 160th Ave SE, Bellevue, WA 98008

can replace or delete portions of the assets, divert them to unsuitable destinations, or replace them with out-dated/modified versions. The execution of corrupted parts on-board airplane may create error conditions that can present a potential threat to airplane safety.

2.2 Threats to Business

Although detection of corrupted assets in the AADS should mitigate safety impacts and airplane certification delays, a late detection (e.g. after loading parts on airplane LRUs), or detection of false positives, could result in delays that disrupt business. Unwarranted delays might also be induced by jamming the distribution or storage of assets using denial of service attacks. Business threats also emerge when entities repudiate access, release or reception of assets.

In order to ensure airplane safety as well as to protect the business of commercial aerospace businesses, the attacks described above must be prevented or mitigated.

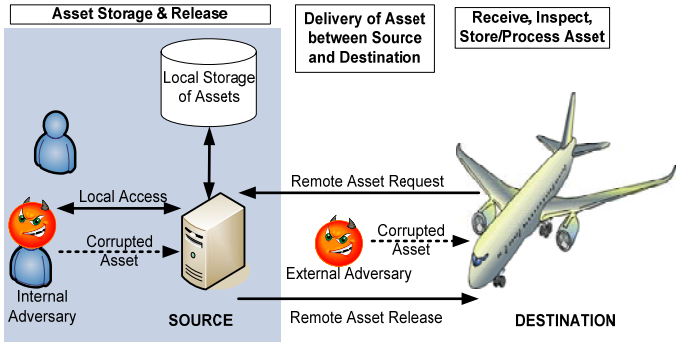

Fig. 2. Distribution of stored assets to airplane in AADS

3. AADS Security Requirements

We are compiling a Common Criteria Protection Profile that asserts the following security primitives suffice to address threats to an AADS.

(a) Authentication: The identities of source and destination must be correct. This ensures that the stored assets are not received from or accessed by an external adversary.

(b) Authorization: The destination (source) must be allowed to access (send) assets only if it possesses the required permissions to do so. This mitigates insider attacks to an extent, while preventing attacks by external adversary.

(c) Integrity: The destination must accept an asset from an authorized, authentic source, only if the asset is the same as that at the source. Further, the stored assets must be protected from any corruption attacks by insiders.

(d) Correct Status: The source (destination) must send (accept) an asset only if it is in release state. This ensures that the distributed asset is safety-assured.

(e) Correct Destination: The destination must accept an asset only if it is the intended destination. This requirement prevents the acceptance of diverted assets.

(f) Correct Version: The destination must accept an asset only if it is the latest version, thereby preventing the acceptance of replayed assets.

(g) Availability: In order to maintain authenticated and authorized access to assets, the hosts storing the assets and the networks used to deliver the assets need to be protected against well known denial of service attacks.

(h) Accountability: For each stored asset distribution, the authentic identities of the source and destination, distribution time, and other relevant information must be recorded. This prevents repudiation by entities, while also mitigating the threats to safety by deterring insider attacks.

Unfortunately, the above requirements cannot prevent insider attacks. This necessitates adding the following two requirements.

(i) Robust Part Design: In order to ensure safety, the part design must facilitate detection of any corruption during execution, and must be fault-tolerant to such modifications. For example, a fault-tolerant part design would require the adversary to corrupt multiple instances of the software.

(j) Early Detection: Corrupted assets must be detected as early as possible. In order to alleviate business disruptions these assets must be ideally detected before being distributed to the airplane. However, this in turn depends on the correctness of the airplane configuration report.

4. Research Directions

We aim at completing the draft Protection Profile soon and fostering its adoption as an industry standard. For interoperability, cost-effectiveness, and rapid development and adaptation, Boeing software development makes significant strategic use of open source software components. The security and software engineering issues identified for AADS generalize to other airplane infrastructure systems. For all of these, we see the need for research in areas including:

(a) Appropriate quality metrics and uses of open source software

(b) Development of sufficient evaluation and certification criteria

(c) Recommendations for selection of appropriate evaluation assurance levels

(d) Use of formal methods for high-quality specifications and high-assurance verification

(e) Requirements for adequately assured public key infrastructure and web service components

(f) Integration of airplane software with air traffic management systems

5. References

[1] ARINC Report 665: Loadable Software Standards, ARINC, August 2005.

[2] G. Bird, M. Christensen, D. Lutz, P. Scandura, Use of integrated vehicle health management in the field of commercial aviation, NASA ISHEM Forum, 2005.

[3] ARINC Report 666A: Electronic Distribution of Software, ARINC, May 2005.

[4] DO-178B: Software Considerations in Airborne Systems and Equipment Certification, Radio Technical Commission for Aeronautics (RTCA), December 1, 1992.