Security Annex Update

May 15, 2018

Dave Gluch

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-0634

Security Annex

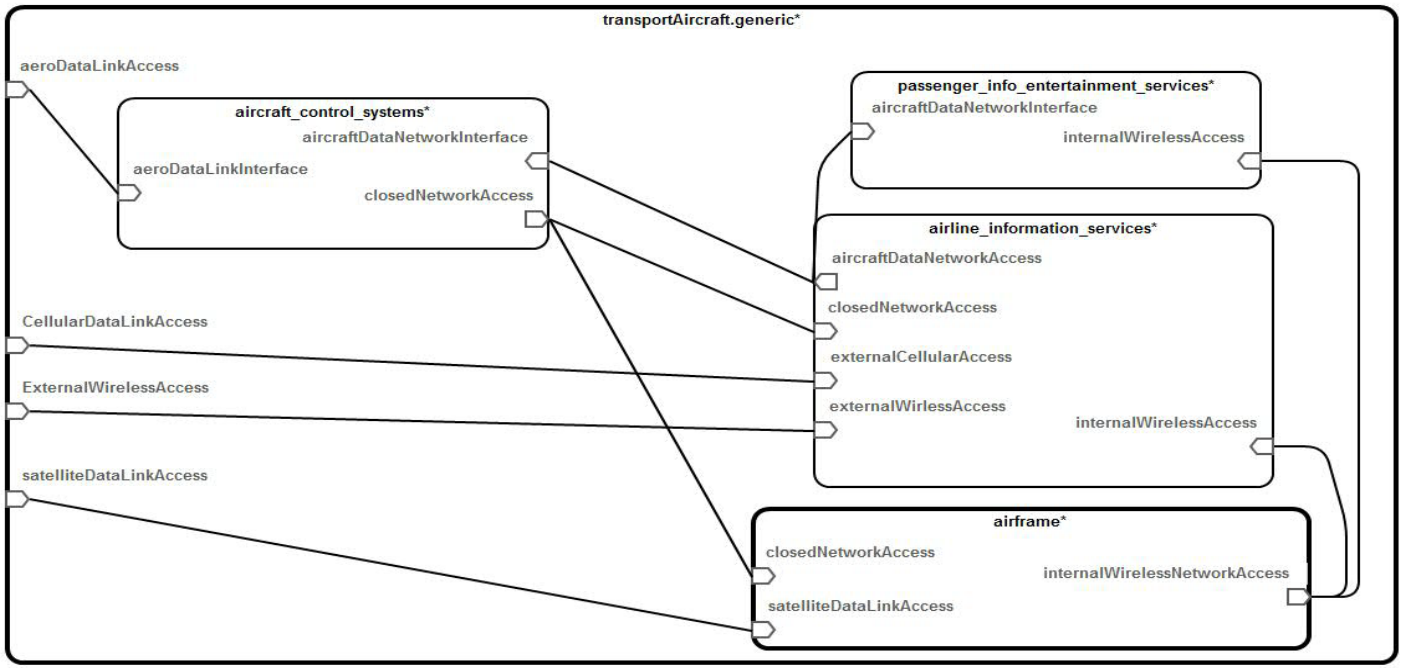
The AADL security annex provides guidance and support for security modeling and analysis throughout the system lifecycle.

- Policies and Requirements
  - Documentation
  - Verification
- Protections
  - Access Control and Protection
  - Information/Data Protection
  - Action/Command Protection
- Architectures
  - Specialized architectures
  - Secure kernels
- Vulnerabilities
- Threats/Attacks

E-Enabled Transport Aircraft

Aircraft Domains

- Aircraft Control (AC)
- Airline Information Services (AIS)
- Passenger Information & Entertainment Services (PIES)



Security Policies and Requirements Documentation

Documentation of policies, requirements.

- ReqSpec capabilities of the ALISA framework
- Naming convention to distinguish between policies and requirements

TransportAircraftSecuirtyPolicies.reqspec

AircraftControlSecurityReqs.reqspec

Security policies considered as general statements (rules) about security attributes of a system

Security requirements considered statements that define the functions and capabilities that must exist to provide security

Examples – Security Policies

File: *TransportAircraftSecuirtyPolicies.reqspec*

```
system requirements TransportAircraftSystemSecurityPolicies : "System-Wide Security Policies"
for TransportAircraftSystem_Generic::AirTransportOperationalSystem.multipassenger
[
description "These are the high level (system) security policies for the Aircraft."
requirement Secuirty:  "System Security must be provided"
[
description "Security protections that meets FAA aircraft security and
flight worthiness certification standards must be provided ."
]
requirement MasterSecurityPolicy: "A Master System Security Policy must be developed and
certified."
[
description "A master system security policy document must be developed and certified by all of the
agencies and organizations involved in flight certification of the aircraft."
]
requirement  AccessControlPolicy: "Security Controlled Access to all Aircraft Systems and Resources
must be provided."
[
description "Access to all Aircraft operational and maintenance Systems and Resources shall be
permitted
only by authorized personnel."
development stakeholder DevelopmentTeam.PrincipalEngineer DevelopmentTeam.SecurityEngineer
]
```

Examples – Security Requirements

File: *AircraftControlSecurityReqs.reqspec*

```
system requirements securityReqs for AircraftControl_pkg::aircraftControl.basic
[
requirement aircraftSystemsInformationSecurity: "Aircraft Systems Information
Security/Protection (ASISP) must be provided"
[
description "All aircraft control and flight information systems must have security
protection to ensure confidentiality, integrity, and availability."
]
requirement securityAccessReq:  "Access to all aircraft data must be only by authorized and
authenticated entities"
[
description "All aircraft operational and performance data systems must have security
protection to ensure access only by authorized and authenticated entities."
]
requirement communicationProtectionReq0: "All external and internal communications relating
to aircraft control and operation must be secure."
[
description "All aircraft communication systems must have security protection to ensure
access only by authorized and authenticated entities."
]
requirement  communicationProtectionReq1: "All aircraft external and internal communication
for aircraft control and flight operations must employ encryption algorithms"
  [
   description "All aircraft external and internal communication for aircraft control and
flight operations must employ encryption algorithms that meet or exceed the standards
defined in NIST publication FIPS 140-2 or any superseding document that have been released
for use."
  ]
```

Security Policies and Requirements Verification

Verification and Assurance

TransportAircraftSecurityVerificationPlan.verify

TransportAircraftSecuityVerifyMethods.methodregistry

TransportAircraftSecurityAssuranceCases.alisa

```
verification methods TransportAircraftSecurityVerifyMethods : "These are
the top-level methods.
[
method ReviewMasterSecurityPolicy (document:string) boolean: "A formal
document inspection and review"
[
manual FormalReview
description “Formal Team review of policy documentation."
]
]
```

Security Protections – Access Control and Protection

Authentication Protocols

```
authentication_type_access: enumeration (NoValue, single_password, smart_card,
ip_addr, two_factor, multi_layered, bio_metric) applies to (all);
--
--  two_factor is a subset of multi_layered but is included since it is a
prevalent multi-layered type.
--
-- The NoValue entry is used as the default in the Resolute built-in property
function,
-- i.e., property (namedelement, property, default value* )
--
authentication_protocol: enumeration (NoValue, cert_services, EAP, PAP, SPAP,
CHAP, MS_CHAP, Radius, IAS, Kerberos, SSL, NTLM)  applies to (all);
```

Authorization Protocols

- establish access rights and actions of an authenticated entity

Information/Data Protection - Security Classifications

United States DoD and DOE classifications

```
        Security_Levels_US: type enumeration (NoValue, SSBI, SCI, SAP, Top_Secret,
Secret, Public_Trust, Confidential, Controlled_Unclassified, Unclassified,
Q_Clearance, L_Clearance);
--
-- Security level property (assumes only one level assigned and encompasses both
-- information classification as well as security clearance.
--
        Security_Level_US: Security_Properties::Security_Levels_US
        applies to (all);
--
-- The secondary security level is provided in the event of multiple clearances
-- (e.g. a clearance from two different agencies.)
--
-- No assumption is made about the relationship between the Security_Level_US
property and the Secondary_Security_Level_US.
--
-- NOTE: the resolute library does not include checks for secondary security levels.
--
        Secondary_Security_Level_US:Security_Properties::Security_Levels_US
        applies to (all);
```

Information/Data Protection - Security Analysis

Resolute security functions and claims in the Resolute annex library *Security_Resolute_Lib*

- basic set for use
- exemplars for users to develop additional functions and claims

```
has_top_secret_security (cp: component) <=
  ** " Component " cp " has Top Secret security level or clearance" **
 property (cp, Security_Properties::Security_Level_US, NoValue) = topSecret

all_subcomponents_have_secret_security(cp: component) <=
** "all subcomponents of component " cp " have secret security level." **
        forall (p: subcomponents(s)).has_secret_security (p)
-- only checks direct subcomponents of the instance; does not include
subcomponents of subcomponents

--
```

Information/Data Protection – Cryptography

From Security_Properties property set

```
        encryption: aadlboolean applies to (all);

encryption_scheme : Security_Properties::encryption_type applies
to (all);
        encryption_type : type record
(
encryption_form : enumeration (symmetric, asymmetric, hybrid,
                    authenticated_encryption, no_encryption, AEAD);

algorithm : enumeration (tripledes, des, rsa, blowfish, twofish,
aes, D_H, clear);
        private_key: aadlstring;  -- maybe better as an integer?
        public_key: aadlstring;
        single_key: aadlstring;
        authentication_type: enumeration (EtM, MtE, E_and_M, AEAD);
        MAC_key : aadlstring;
```

Information/Data Protection Encryption Analysis

Example Claims from Security_Resolute_Lib

```
has_encryption (aadl1: aadl) <=
    ** "AADL element" aadl1 "has encryption" **
    has_property(aadl1, Security_Properties::encryption)
--
   has_an_encryption_scheme (aadl1: aadl) <=
   ** "AADL element "  aadl1 " has a value for encryption_scheme"
**
   has_property (aadl1,Security_Properties::encryption_scheme)

all_contained_buses_have_encryption (cp:component) <=
       ** "all buses contained in component " cp " have encryption"
**
       -- the exists claim ensures that there is one bus in cp;
      -- without this, the claim is true if there are no buses
       (exists(bx: bus).contained(bx,cp) and
       (forall( bt: bus).contained(bt,cp) and (has_encryption(bt) or
has_an_encryption_scheme(bt)))
```

Information/Data Protection
Protected Containment and Access

Secure Virtualization

- AADL core modeling capabilities
- ARINC 653

Data access control

- Authentication
- Authorization

Action/Command Protection

Model, assess, and assure access control of execution of actions/commands including

- Security kernels (e.g. seL4)
- Operating system security controls
- Specialized operating systems
- Protected Containment
- Virtual Trusted Execution Environments
- External Actions
- Virtual machines, and partitions

Security Architectures (Modeling)

Specialized Architectures

- AADL core modeling capabilities and ARINC 653
  - temporal and spatial isolation
  - potentially specialized security constructs

Secure kernels

- AADL core modeling capabilities & ARINC 653

Vulnerabilities – Documenting1

AADL Property

```
Vulnerabilities: list of record
  (
   Name : aadlstring; -- short identification phrase for the vulnerability
   Description : aadlstring;      -- description of the vulnerability
   CrossReference : aadlstring;   -- cross reference to an external document
   Phases : list of aadlstring;   -- operational phases in which the
                                  vulnerability may be exploited
   Environment : aadlstring;      -- description of operational environment
   Threat : aadlstring;           -- description of the circumstances under
                                  which the vulnerability may be exploited
  Loss : aadlstring; -- description of the loss that may result
Risk : aadlstring;  -- description of risk
Severity : EMV2::SeverityRange ; -- actual risk as severity
   Likelihood : EMV2::LikelihoodLabels;    -- actual risk as
likelihood/probability
   Probability: EMV2::ProbabilityRange;    -- probability of a exploitation
(i.e. realization of loss)
```

Vulnerabilities – Documenting and Analysis

Vulnerabilities Property (continued)

```
AcceptableSeverity : EMV2::SeverityRange; -- acceptable risk as severity
AcceptableLikelihood : EMV2::LikelihoodLabels; -- acceptable risk as
                                               likelihood/probability
DevelopmentAssuranceLevel : EMV2::DALLabels;          -- level of rigor in
                                               development
 VerificationMethod : aadlstring; -- verification method to address the
                                     vulnerability or threat
 SecurityReport : aadlstring;     -- analysis/assessment of hazard
 Comment : aadlstring;

       )  applies to (all);
```

Analyzing Vulnerabilities

- EMV2
- Integrated security and safety analysis

Threat/Attack Modeling

Capture and analyze threat/attack models including:

- Threat models
- Attack/attacker models
- Attack surface models
- Attack trees
- Chain of events models
- Attack patterns
- Denial of Service

May not be included?

Summary

Security Property Set

Security Resolute Library

Test Cases for Claims

ALISA Exemplar Files

E-Enable Aircraft Models