SECURING THE SKIES:

IN REQUIREMENTS
WE TRUST

Bashar Nuseibeh  |  Charles B. Haley  |  Craig Foster

SEPTEMBER 2009

Presenter: Or Shachar or12@cs
Bashar Nuseibeh, PhD

The Open University (UK)

- Software Engineering chief scientist
  The Irish Engineering Research Center
- Head of software engineering lab
  Imperial College London
- Fellow:
  Automated Software Engineering.
  British Computer Society
  Institution of Engineering and Technology
Charles B Haley, PhD
The Open University (UK)

- **Teacher**
  Many areas of CS
- **Researcher**
  Representation of Security Requirements
  Argumentation
  Digital Forensics*
Craig Foster, Mr.
National Air Traffic Services (NATS)

- Navigation & Surveillance Researcher
- Project Manager

*Eurocontrol CASCADE program*
Complex system security

- **Security** is much about understanding the context in which the system operates as it is about the systems themselves.

- A sociotechnical system compromises hardware, software and people.

- It is users and their assets that are harmed from an attack on the system.
Complex system security

Organizations must look beyond the system to examine:

- WHAT they are trying to protect?
- WHY they are trying to protect it?
- CONSEQUENCES of inadequate protection

(Security) requirement engineering considers those questions and elicit the

SECURITY REQUIREMENTS
Security Requirements

Part of security requirement engineer challenges:

- Identify stakeholders
- Wider problem scoping
- Representation of security requirement
- Requirement analysis

Like other requirements, security req. should not be too general nor overly specific
Framework for finding the right SR

ID System context & Functional Requirements

ID Security Goals

ID Security Requirements

Construct Satisfaction Argument

Quality / Business goals

Infeasible / Secondary SG needed

Feasible

Infeasible

(REVISE)

Feasible

Infeasible

(REVISE)

Feasible

Infeasible
Air Traffic Control (ATC) – need of exact position and altitude of aircraft at any given moment.

<table>
<thead>
<tr>
<th>THE OLD METHOD</th>
<th>THE SUGGESTED METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground RADAR (Active surveillance)</td>
<td>Aircrafts’ GPS (Passive surveillance)</td>
</tr>
</tbody>
</table>

- Aircraft equipment independent
- Expensive

- Aircraft equipment dependant
- Advanced, cost saving

What are the suggested method security requirement?
Step 1: Produce Functional Requirement

ID system goals and quality goals

Construct system context

ID functional requirement

To step 2

Feasible

(REVISE)

Infeasible

OK

Infeasible
Step 1: In action

System goal was already given:

“Provide safe and efficient air traffic management.”

Existing equipment:

**ADS-B equipment**

⇒ FR: provide position of aircraft
Step 1: Example of system context
Step 2: Produce Security Goals

1. Select security control principle
2. Elicit / revise asset

ID security goals

Validate goals against assets, threats

Functional requirement and system context from step 1

To step 3

Start Over (step 1)
Step 2: Example Assets Mapping
Step 2: In action

Security principles:

Confidentiality | Integrity | Availability

⇒ Formal Threads representation

i.e: T3: {~correct, airplanes’ position, lost revenue due to increased separation}

Security goals:

- Have correct positions
- Report positions as often as needed
Step 3: Produce Security Requirements

Think of constraints to place on functional req. to that will satisfy the goals.
[Very immediate]
Step 3: in action

Security goals:

- **SG1**: Have correct positions
- **SG2**: Report positions as often as needed

Security requirements:

[On FR1: Provide position of aircraft]

- **SR1**: Positions shall be accurate
- **SR2**: Positions shall be timely
**Step 4: Validate Satisfaction Ability**

- **Determine secondary security goal**
  - Feasible
  - Infeasible
  - OK (REVISE)

- **Revise and verify system context**
  - Feasible
  - Infeasible
  - OK (REVISE)

- **Validate SRs satisfy goals**
  - OK
  - Start Over (step 1)

- **Security requirements (step 3)**

- **Outer argument**
- **Inner arguments**
Step 4: In action

Outer argument:

Aircraft gets accurate GPS info → Position sent to ATC
Step 4: List of terms for outer argument

- Aircraft gets accurate GPS info
- Aircraft sends accurate position
- Receiver sends position
- Position sent to ATC
Step 4: Assumption test example

Aircraft gets accurate GPS info

Aircraft sends accurate position

Warrants:
- Calculations are accurate

Grounds:
- Received GPS positions are accurate

Claim:
- Accurate positions are transmitted

Rebuttals:
- R1.1: Aircraft’s GPS sabotaged (SR2)
- R1.2: ADS-B transmitter sabotaged (SR2)
- R1.3: Aircrew transmits wrong position/ID (SR1)
**Step 4: Assumption test example**

Aircraft gets accurate GPS info

Aircraft sends accurate position

**Warrants**

Calculations are accurate

**Grounds**

Received GPS positions are accurate

**Claim**

Accurate positions are transmitted

**Rebuttals**

- R1.1: Aircraft’s GPS sabotaged (SR2)
- R1.2: ADS-B transmitter sabotaged (SR2)
- R1.3: Aircrew transmits wrong position/ID (SR1)

Possible terrorist attack – must be addressed!
Step 4: Assumption test example

Aircraft gets accurate GPS info

Aircraft sends accurate position

Possible solution: Multilateration
Lesson Learned

- Use domain experts
- Use domain non-experts (Remember D. Berry)
- Scope the problem (WIDER than you might think)
- Iterate to mitigate
- Formalize but argue informally too.

security is much about being persuaded "beyond reasonable doubt" that a system is secure than it is about a proof of security, whatever that means
Summary and Discussion

Better Req. $\rightarrow$ Better system

Better security Req. $\rightarrow$ Better secured system

- **Powerful tool** – Intelligent requirement
  
  Proof of security

- **Security is more and more important**

- **“Secure”** – against lost of assets

  against possible attacks (Thompson)

- Learn more: Security principles, “legally secured”