Security Minded Verification & Validation for Future Space Systems via Reference Architectures

Matthew Bradbury

Cyber Security Centre, WMG, University of Warwick

6th June 2019

Introduction

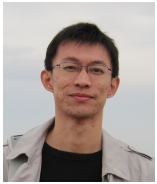
1. Summary of the FAIR-SPACE Hub
2. Security Minded V&V
3. Using Reference Architectures for Attack Surface Analysis
4. Lessons Learnt from Connected Autonomous Vehicles

The Team

WMG
THE UNIVERSITY OF WARWICK


Carsten Maple


Hu Yuan


Chen Gu


Ugur Ilker Atamaca


Chronis Kapalidis

UNIVERSITY OF LIVERPOOL


Michael Fisher


Clare Dixon


Louise Dennis


Marie Farrell

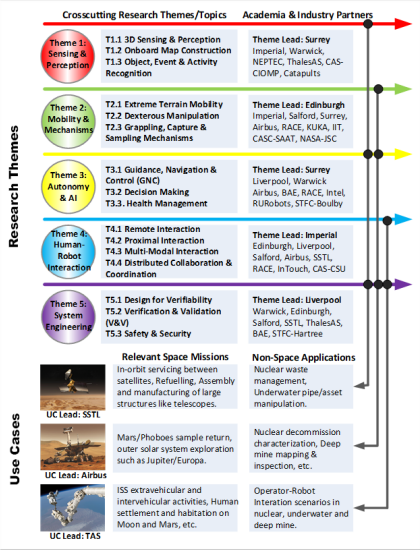

Matt Webster

FAIR-SPACE Hub I

The aim of the Future AI and Robotics for Space (FAIR-SPACE) Hub is to **go beyond the state-of-the-art** in robotic sensing and perception, mobility and manipulation, on-board and on-ground autonomous capabilities, and human-robot interaction, and to **enable long-lived robotic operations in space**.

- To merge the best available off-the-shelf hardware/software solutions with trail-blazing innovations, new standards and frameworks, leading to a constellation of **space RAI prototypes and tools**.
- To accelerate the prototyping of autonomous systems in a scalable way, where the innovations and methodologies developed can be **rapidly adopted by the space industry**.

FAIR-SPACE Hub II



- Five cross-cutting research themes underpinning major industry-led challenges.
- Each research theme addresses a set of scientific topics and objectives through collaborative projects between academia & industry.
- Outputs from selected themes are coherently combined within industry-defined use cases to demonstrate new knowledge and technologies.

FAIR-SPACE Hub III

Use-Cases:

Orbital: robots for repairing satellites, assembling large space telescopes, manufacturing in space, removal of space junk

Planetary: for surveying, observation, extraction of resources, and deploying infrastructure for human arrival and habitation

Human-robot: will target astronauts-robot interoperability aboard the International Space Station or for the future Moon Village.

Security Minded Verification & Validation

Verification and Validation (V&V)

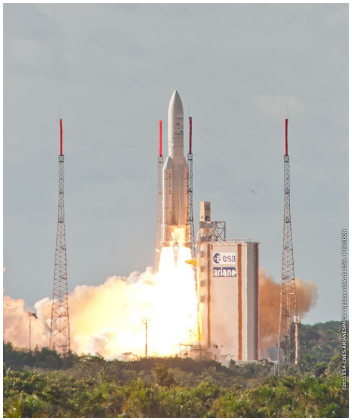
- Verification: Does the system meet the specification?
- Validation: Does the system meet the needs of the user?

In other words:

- Verification: Are you building the thing right?
- Validation: Are you building the right thing?

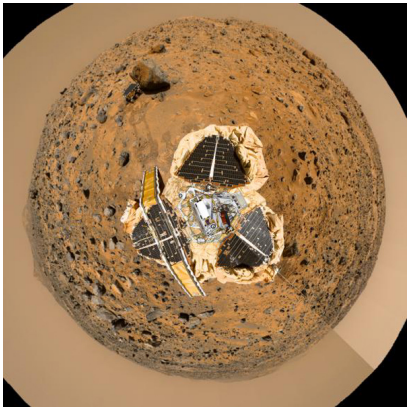
— Barry Boehm, Software Engineering Economics, 1981

Ariane 5 Flight 501 (04/06/1996)



30 seconds after launch the rocket deviated 90 degrees from flight path. Destroyed by aerodynamic stresses. Caused by overflow due to casting a 64-bit float to a 16-bit integer. Backups failed for the same reason.

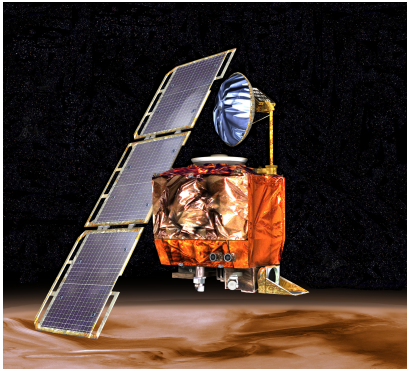
Mars Pathfinder (04/07/1997)



(a) © NASA/JPL

Priority inversion in scheduling system led to system resets. A program was deployed that allowed the problem to be fixed.

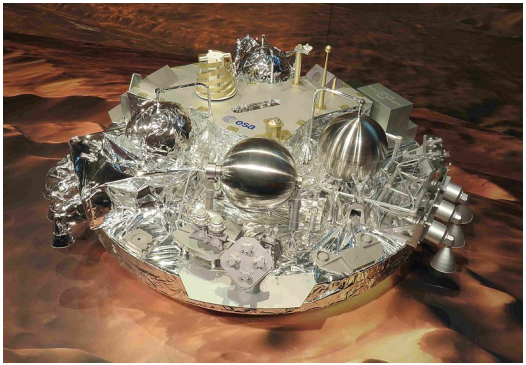
Mars Climate Orbiter (23/09/1999)



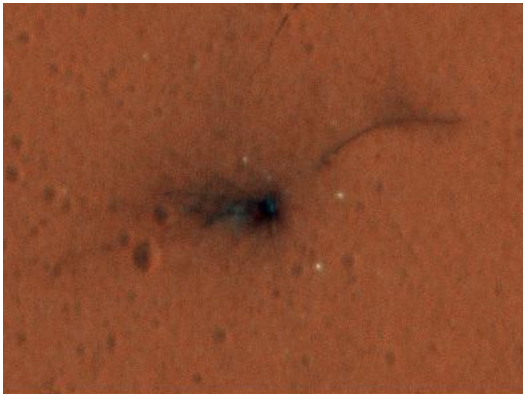
(a) © NASA/JPL

Communications lost during orbital insertion due to mismatch between non-SI units of pound-force second instead of SI units of newton-seconds.

Schiaparelli EDM (19/10/2016)



(a) Schiaparelli Lander Model at ESOC



(b) © NASA/JPL-Caltech/Univ. of Arizona

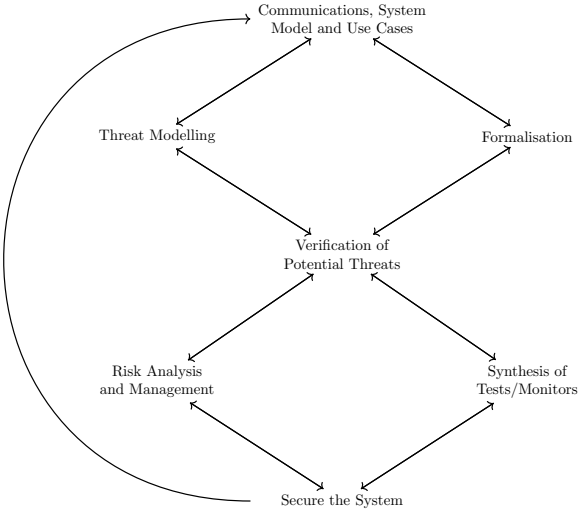
The inertial measurement unit became saturated and led to a negative altitude reading.

V&V Considerations

- These systems are large and complex — where should efforts be focused?
- Can't prove everything (usually)
- Machine learning is especially challenging

Security Minded V&V

- Which aspects of the system are going to be important to protect cyber security?
- How do we identify these areas?
- What if we are dealing with systems of system?
- Which security properties are important and how do we formalise them?



Which security properties are important? — Ask the stakeholders!

`https://fairspacehub.org/s/workshop_report_space_security_scoping.pdf`

Reference Architecture for Attack Surface Analysis

Requirements for Attack Surface Analysis

- To understand how to attack a system we need to know:
  - Who will perform the attack,
  - What their capabilities are,
  - Why they are attacking the system,
  - Where their attacks originate from,
  - How their attacks reach the target
- We have to focus on specific aspects of the system (the two use cases) to constrain our analysis to a reasonable size
- But we need a high-level overview of the rest of the ecosystem to understand how the use cases can be attacked

Adversaries

Threat Actor	Motivation	Resources	Access	Knowledge
Nation-States	State rivalry, Geopolitical	Extensive	Remote coordinated	Easy to develop knowledge
Commercial Competitors	Corporate espionage, Financial gain	Company size-related	Remote coordinated	High to moderate knowledge
Organised Crime	Financial gain	Moderate availability	Remote coordinated	Moderate knowledge
Amateur Cracker	Personal satisfaction	Minimal	Remote	Limited
Insiders	Discontent, Financial	Minimal	Privileged access	Internal Knowledge

Adversary Presence

Presence of the adversary is specified relative to the target of the attack

- **Internal**: Has physical access to the internals of the target
- **Local**: Adversary has physical access to the external surface of the target
- **Semi-local**: Adversary is physically nearby the target (e.g., within direct communication or sensor range)
- **Remote**: Adversary is physically far from the target (e.g., access via the internet or satellite network)

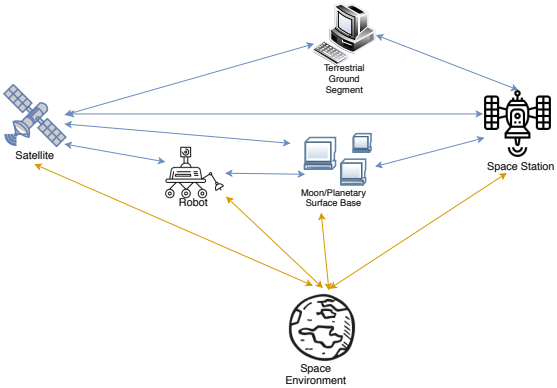
The presence impacts the kinds of threats an adversary can carry out

Threat Modelling

- Many different approaches to modelling threats
- Want to think about "what can go wrong in the system?"
- Not all possible threats will be applicable, depending on the adversary, its capabilities and its presence
- How are threats classified (CIA, STRIDE, . . .)?

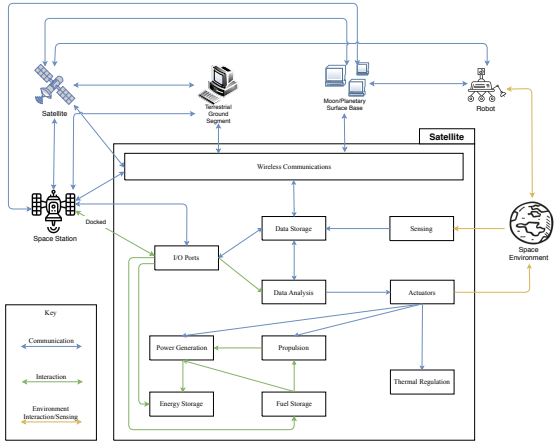
High Level Reference Architecture

- Need to understand high level **Functionality** and **Interactions** of the system
- Reference Architectures are good ways to encapsulate this information
  - Functional Components in boxes — What does the component *do*, what is its purpose?
  - Interactions given by the connections between components
- 5 key sub-architectures on right, focusing on the in orbit and planetary aspects for now



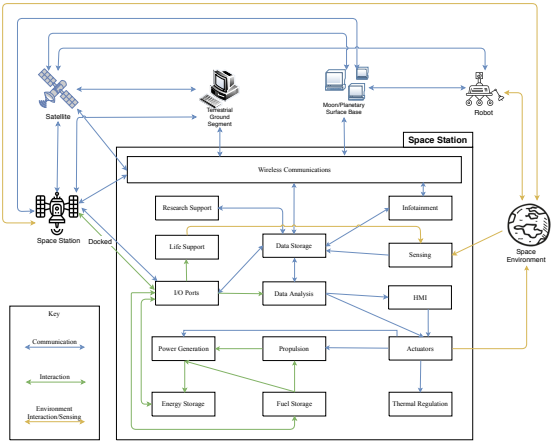
Satellite

- Focus on **high level** functional activities the satellite performs
- Not all satellites need to provide all of this functionality
- Satellites are capable of providing this functionality **in general**
- Arrows that cross the boundary of the devices are of interest to analyse the attack surface
- Need to think about how an attack research internal components

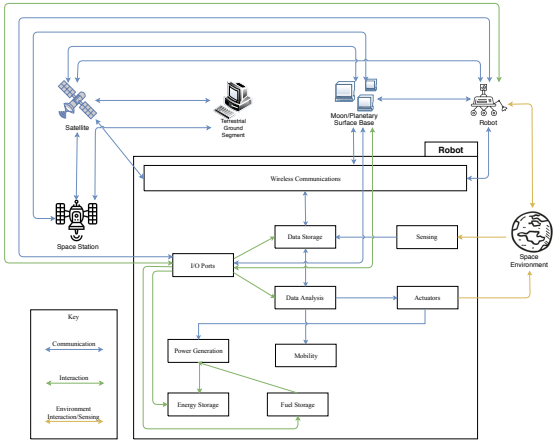


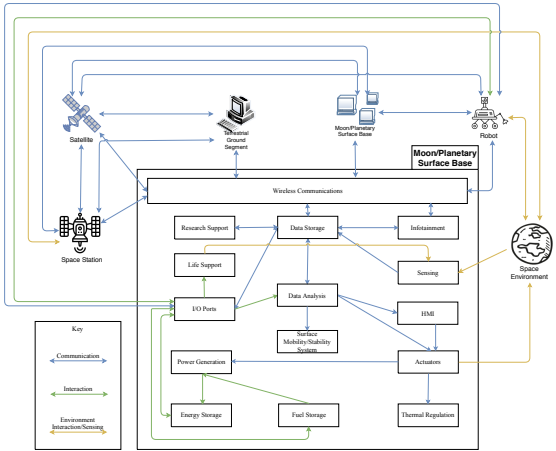
Space Station

- Very similar to Satellite
- Differences in functionality caused by the need to support human life
- Sometimes it is useful to separate functionality that overlaps due to its importance (e.g., propulsion could be through of as an actuator)
- It is useful to include the other sub-architectures to trace attacks through the system in one diagram



Robot and Surface Base





Example Analysis

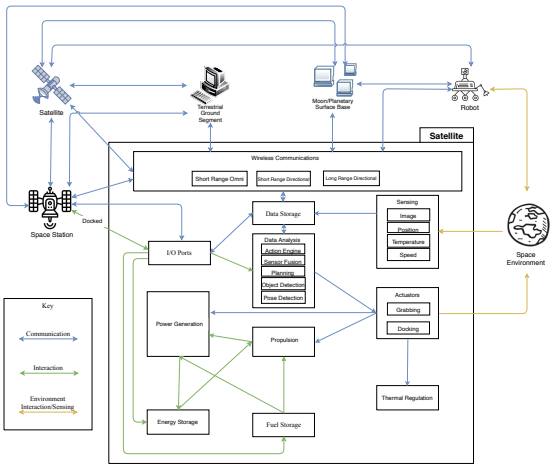
- Scenario: Automated Exploration of the Planetary Surface
- Adversary: Commercial Competitors
- Objective: Reduce effectiveness of competitors robots
- Presence: Semi-Local (via robots deployed on the surface) and Remote (via internet and satellite network)

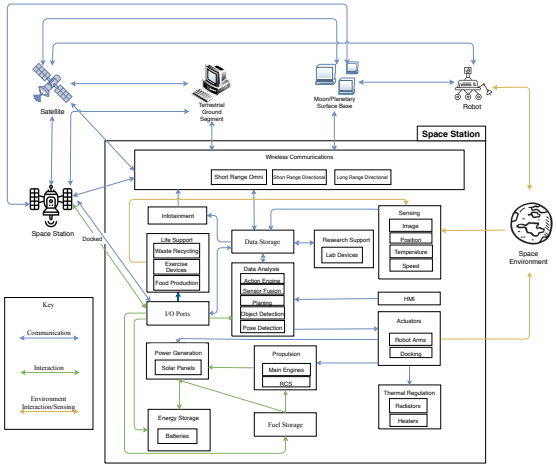
Requirements specify that the autonomous robots should:

- Minimise: Energy usage, task duration, navigation risk
- Optimise: Attitude (e.g., solar panels)

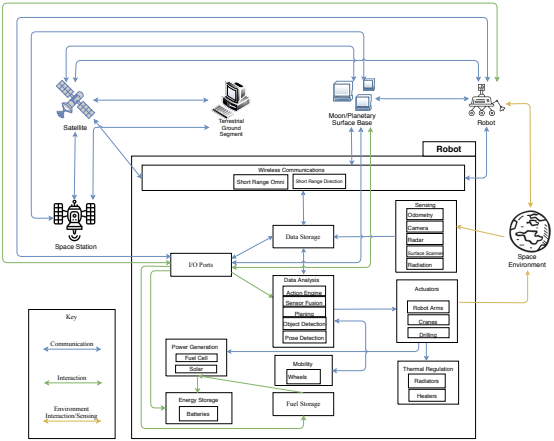
These are aspects that the adversary will attempt to worsen

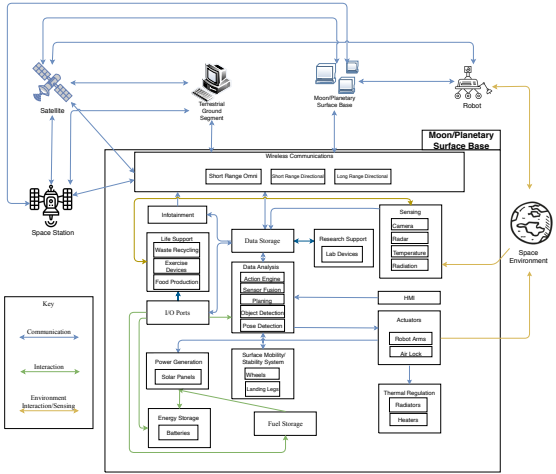
Instantiation of Satellite and Space Station





Instantiation of Robot and Surface Base





Identified Threats for Scenario

- Semi-local
  - Denial of Service: Jam Sensors, DoS communications (disrupt Robot-to-robot communications)
  - Information Disclosure: Position Tracking
  - Repudiation: Pretend to collaborate but lie about work done for collaboration
  - Spoofing: Pretend to be a robot belonging to a different organisation
- Remote
  - Tampering: Maps Poisoning
  - Denial of Service: Prevent relay satellites from communication between robots and terrestrial stations. Intercept satellite handover

Future Work for the Reference Architecture

- To investigate cyber-physical attacks we need a **Physical** viewpoint
  - Throw dust on solar panels
  - Change environment to prevent other robots recognising it or to hide an interesting find
- Need to consider aspects of the ecosystem based on Earth, such as the Terrestrial Ground Stations, Relay sites, etc.
- How to design the system to be resilient to these attacks? How to implement countermeasures against these attacks?
- Risk analysis to prioritise the most likely and impactful threats.

Lessons Learnt from Connected Autonomous Vehicles

Similarities between CAVs and Space Systems

Current similarities:

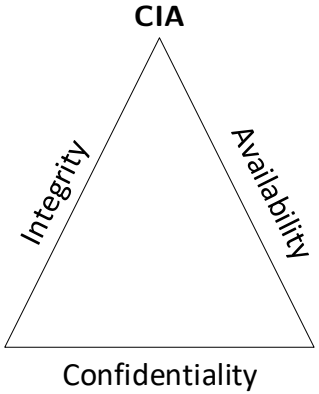
- Limited computational power
- Resource constrained (fuel, electricity)

Future similarities:

- Autonomous systems in CAVs, autonomy expected in future space systems
- CAVs expected to operate in more hazardous environments in future (e.g., high radiation environments)

Is Encryption Always Needed? I

It depends on what security properties you want.



STRIDE

- Authenticity
- Integrity
- Non-repudiation
- Confidentiality
- Availability
- Authorisation

Is Encryption Always Needed? II

- **Authenticity** — Need a way to prove the identity of the sender of the message.
- **Integrity** — Need a way to validate the received contents is the same as the sent contents.
- **Non-repudiation** — Need a way to prove the identity of the sender of the message.
- **Confidentiality** — Encryption.
- Availability — Protocol and operation considerations.
- Authorisation — Need a way to prove the identity of the sender of the message and check what permissions they have.

Is Encryption Always Needed? III

Security Property	Cryptographic Primitive			
	Hash	MAC	Digital Signature	Encryption
Integrity	✓	✓	✓	✗
Authenticity	✗	✓	✓	✗
Non-repudiation	✗	✗	✓	✗
Confidentiality	✗	✗	✗	✓

MAC cannot provide non-repudiation as at least two entities have the secret key.

Conclusions

- Need a high assurance of the correct and secure operation of space systems
- Formal verification is an important component of this
- The system of systems will be large
- Hard to know which aspects of the system to focus on verifying
- Reference Architectures are useful tools to identify points of attacks in space systems
- Reference Architectures also useful for understanding how an attack propagates through the system
- Experiences from automotive (i.e., CAVs) will be useful for future space systems

Thank you for listening.

Any questions?