Security-Informed Safety

Dr Ayman El Hajjar

CyBOK Mapping

The lecture maps to the following CyBOK Knowledge Areas:

Full Coverage:

- **Infrastructure Security → Cyber Physical Systems**

Partly Covered:

- **Human, Organisational & Regulatory Aspects → Risk Management & Governance**
- **Attacks & Defences → Adversarial Behaviours**

Outline

The Convergence Problem

The Security-Informed Safety Concept

From Security Breach to Safety Hazard

Building Assurance: Safety Cases and CAE

Integrating Security into Safety Risk Assessment

Common Mistakes, Challenges, and Regulatory Drivers

The Convergence Problem

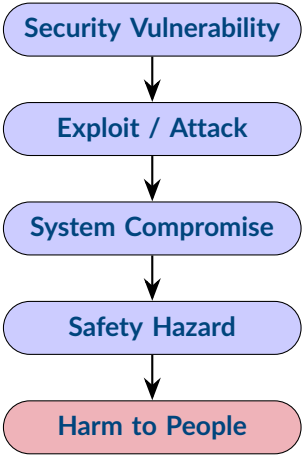
The Convergence Problem

- Security protects assets from **malicious interference**
- Safety protects people from **harm** caused by system behaviour
- In connected systems, a **security breach** can directly cause a **safety hazard**

The problem

How can we claim a system is **safe** if attackers can compromise the systems it depends on?

The Causal Chain: Breach to Hazard



e.g., unauthenticated MQTT broker

e.g., attacker publishes false sensor data

e.g., BMS displays wrong temperature

e.g., HVAC shuts down in occupied building

e.g., heat exposure, respiratory distress

The Security-Informed Safety Concept

Attacks Are Not Failures

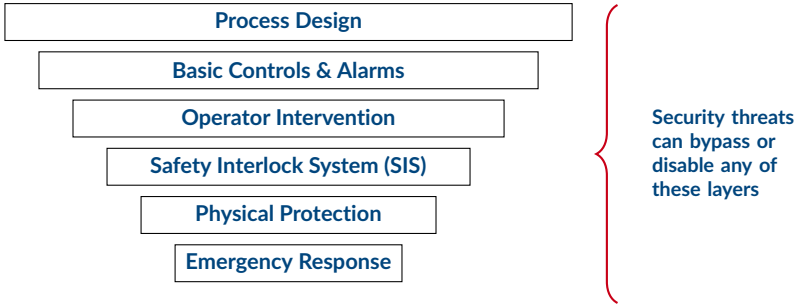
Core Problem: Safety engineering assumes failures are independent and random. Attackers violate both assumptions.

	Accidental Failure	Deliberate Attack
Cause	Wear, defect, environment	Intelligent adversary
Distribution	Random, statistically modelled	Targeted, strategic
Redundancy helps?	Yes (independent failures)	Not necessarily (common mode)
Adapts to defences?	No	Yes
Worst case	Single point of failure	Coordinated multi-point attack

Protections against **non-malicious failures** are not enough against **strategic attackers**.[1]

[1] CyBOK CPS Security, Section 21.1.2

Layers of Protection and Where Security Fits



If attackers can target the **safety mechanisms** themselves, how do we justify that a system is **safe**?

Security-Informed Safety in Practice

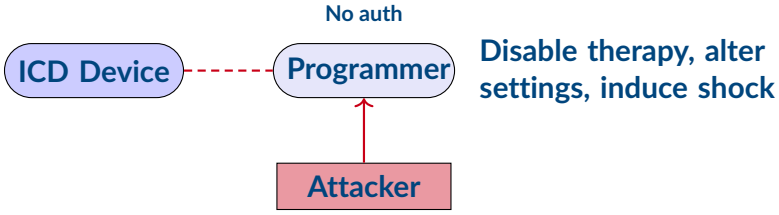
Security-Informed Safety in Practice Security-informed safety requires three actions: treat attacks as hazards, include security evidence, and maintain the safety case as threats evolve.

1. **Treat attacks as a hazard class**
   - Security threats appear alongside hardware failures, software defects, and human error in the hazard register
2. **Include security evidence in the safety case**
   - Pen tests, access logs, encryption verification become safety evidence
   - Missing security evidence means hidden assumptions
3. **Maintain the safety case as threats evolve**
   - New vulnerabilities invalidate existing security evidence
   - Security landscape evolves; the safety case must too

From Security Breach to Safety Hazard

Case Study 1: Implantable Cardiac Devices

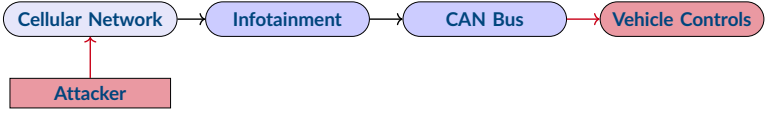
- **System:** ICD with wireless programmer interface, designed with redundancy and fail-safe defibrillation
- **Security finding**[1]: Wireless reprogramming demonstrated with no authentication. Attacker could disable therapy or induce fibrillation
- **Safety impact:** A mechanically sound device became unsafe. Traditional safety mitigations (redundancy, fail-safe) do not address this threat



[1] Halperin et al., 2008

Case Study 2: Jeep Cherokee Remote Exploit

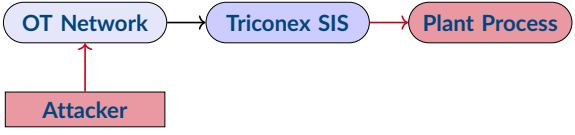
- **System:** 2014 Jeep Cherokee with Uconnect infotainment system
  - Cellular connectivity for navigation and entertainment
  - CAN bus connecting infotainment to vehicle control systems
- **Security finding** [2]**:** Remote exploit via cellular into infotainment unit, pivoted to CAN bus, controlled steering, braking, and transmission
- **Safety impact:** Passed safety certification but missed remote network access as a hazard source. Resulted in 1.4M vehicle recall



[2] Miller & Valasek, 2015

Case Study 3: Triton/TRISIS: When Attackers Target Safety Systems

- **Incident (2017):** Triton malware targeted the Safety Instrumented System at a petrochemical plant in Saudi Arabia
- **What happened:** Attackers accessed the OT network, targeted the Schneider Electric Triconex safety controller, attempted to reprogram the SIS to allow unsafe conditions
- **Significance:**
  - First known malware designed to disable safety protections
  - An attacker controlling the SIS can cause a physical disaster while the control system reports normal operation



What the Case Studies Show

Implantable Cardiac Devices

Safety analysis that excludes **security threats** produces a false assurance of safety.

Jeep Cherokee

Network connectivity created a new **attack surface** that safety testing did not cover.

Triton/TRISIS

Safety systems designed as the final line of defence can themselves be the **target of attack**.

Building Assurance: Safety Cases and CAE

What is a Safety Case?

A **safety case** is a structured argument, supported by evidence, that a system is **acceptably safe** for a given application in a given environment.

- **Used in safety-critical industries:** nuclear, aviation, rail, medical devices, defence
- A safety case is **not** a checklist or a pass/fail test
- A **reasoned argument** that evolves as the system and threats change

"Why should we believe this system is safe?"

Functional Safety Standards

- **IEC 61508:** General functional safety standard for electrical/electronic/programmable systems across industries
- **IEC 61511:** Functional safety for Safety Instrumented Systems (SIS) in process industries
- Both define how to **design**, **validate**, and **maintain** systems against failure. But they assume failures are **accidental** and **random**

Assumption Gap: Traditional functional safety assumes accidental failure, not connected systems facing intelligent adversaries.

Defence-in-Depth: The Underlying Assumption

- **Core assumption:** Failures are **accidental**, **independent**, and **random**
  - Hardware degrades statistically; software bugs appear systematically
  - Redundant systems fail independently, so simultaneous failure is rare
- Therefore **redundancy** works, **testing** prevents failures, **defence-in-depth** holds
- **An intelligent attacker** violates all three: **deliberate**, **targeted**, causes **common-mode failures**

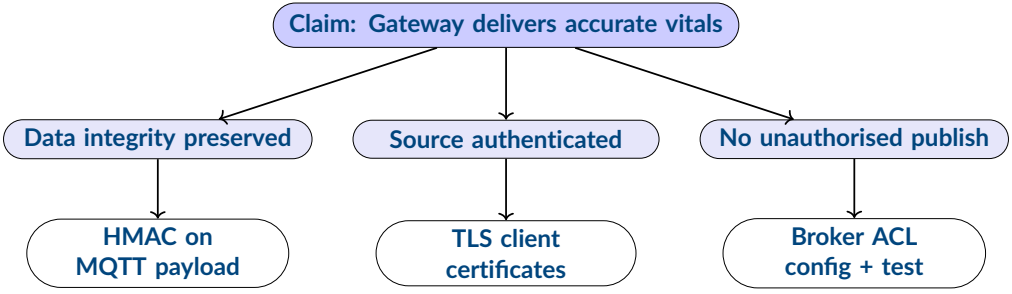
Claims, Arguments, Evidence (CAE)

CAE Structure

CAE structures a safety case as **claims**, **arguments**, and **evidence**.

- **Claim:** A statement about safety that needs demonstrating, decomposed into sub-claims
  - "The patient monitoring gateway does not deliver false vital signs to clinicians"
- **Argument:** Reasoning that connects claims to evidence, explaining **why** evidence is sufficient
  - "All data paths are authenticated and integrity-checked"
- **Evidence:** Concrete artefacts substantiating the argument
  - "TLS 1.3 mutual auth verified by independent pen test "

CAE: A Security-Informed Example



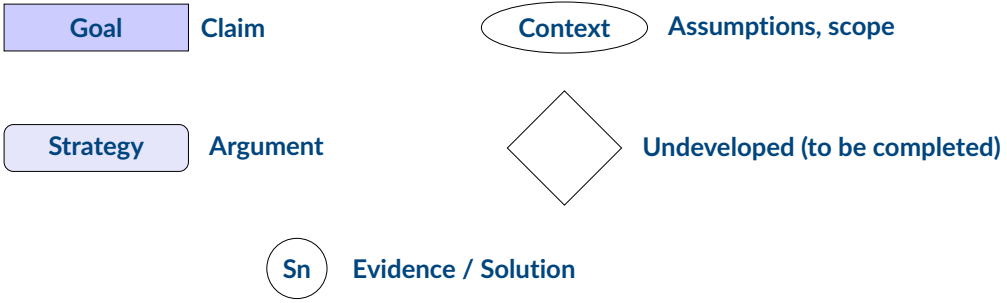
Evidence Gap

If any **evidence link breaks** (e.g., MQTT broker has no ACL), the sub-claim is unsupported and the **top-level safety claim fails**

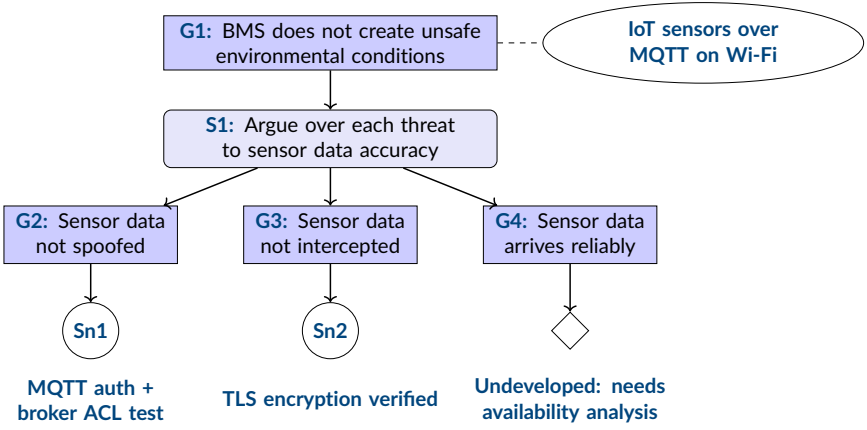
Goal Structuring Notation (GSN)

Goal Structuring Notation

GSN is a graphical notation for representing safety cases. CAE maps directly onto it.



GSN Example: IoT Building Management System



G4 is marked undeveloped , availability analysis is still required before this safety argument is complete

Building a Good GSN Argument

1. **Define the goal:** what safety property must be demonstrated
2. **State the context:** system boundaries, assumptions, and environment
3. **Choose a strategy:** how the goal decomposes into sub-goals
4. **Support with evidence:** provide evidence or decompose further, mark unsupported nodes as undeveloped

Quality checks Every goal is **supported**, reasoning is **explicit**, evidence is **verifiable**, assumptions are **stated**

Why Assurance Breaks Without Security

Patient Monitoring System

Consider a traditional safety case for a patient monitoring system.

1. **Claim:** "Displayed vital signs are accurate"
2. **Traditional evidence:** sensor calibration, hardware redundancy, software validation
3. **Missing:** no argument covers the wireless link. What if it is spoofed? What if an attacker injects false MQTT messages?
4. **Result:** the safety case has an **unstated assumption** that the communication channel is trustworthy

Assurance Gap: Unstated assumptions create exploitable gaps in the safety argument.

Integrating Security into Safety Risk Assessment

Risk

Risk = **Likelihood** $\times$ **Severity**

- **Traditional Safety Risk:** likelihood from failure rates (MTBF), severity in harm to people, mitigated by redundancy and fail-safe design
- **Security Risk:** likelihood depends on attacker capability and opportunity, not statistically predictable, mitigated by access control, encryption, monitoring

Security-Informed Safety Risk Security-informed safety risk combines **hazard sources**, **safety consequences**, and the security controls needed to keep mitigations trustworthy.

The Extended Threat Model

Extended Threat Model

Security-informed safety requires considering all **failure causes**, including malicious attack.

Failure Type	Characteristics	Safety Implication
Random hardware	Statistically modelled	Handled by redundancy
Software defect	Systematic, repeatable	Handled by testing/verification
Human error	Probabilistic, context-dependent	Handled by UI/training
Environmental	External, detectable	Handled by sensors/monitoring
Malicious attack	**Intelligent, adaptive**	**Not handled by above**

Traditional safety covers the first four. **Malicious attack** requires new methods, new evidence, and new expertise

Security-Informed Hazard Analysis

Security-Informed Hazard Analysis Traditional hazard analysis is extended by explicitly including security threats.

1. **Identify safety-critical functions** e.g., display patient vitals, control HVAC
2. **Identify data and control paths** supporting those functions (sensors, networks, gateways, actuators)
3. **For each path:** what if an attacker can read, modify, block, or inject data?
4. **Assess the safety consequence** of each attack using severity categories from safety standards
5. **Identify security controls** that prevent or detect the attack (authentication, encryption, monitoring)
6. **Document in the safety case**, security control claims become part of the assurance argument

Worked Example: Smart Building MQTT Sensor

Step	Traditional Safety	Security-Informed
Critical function	Temperature monitoring for HVAC control	Same
Data path	Sensor → wired → BMS	Sensor → Wi-Fi → MQTT → BMS
Hazard	Sensor drift/failure	Sensor drift/failure **+ spoofed MQTT message**
Consequence	HVAC operates on stale data	HVAC acts on **false** data (attacker-controlled)
Mitigation	Sensor calibration, redundancy	Calibration, redundancy **+ MQTT auth + TLS + ACL**

Security-Safety Shift: The wireless/MQTT path is not just a communication channel; it is part of the hazard analysis.

Applied: 6-Step Analysis on the ICD Case

Step	Action	ICD Application
1	Safety-critical function	Deliver defibrillation therapy when cardiac arrhythmia detected
2	Data/control paths	Sensor $\rightarrow$ ICD logic $\rightarrow$ shock delivery; Wireless programmer $\rightarrow$ ICD configuration
3	Attack possibilities	Attacker reprograms ICD via **unauthenticated** wireless interface: disable therapy, alter thresholds, induce shock
4	Safety consequence	**Patient death** (therapy disabled during arrhythmia) or induced fibrillation
5	Security controls	Mutual authentication, encrypted commands, proximity-based access
6	Safety case entry	Claim: "No unauthorised reprogramming is possible." Evidence: pen test + auth protocol verification

Applying the 6 steps to a real case shows how **security threats** integrate into the **safety analysis** framework

Common Mistakes, Challenges, and Regulatory Drivers

Common Mistakes in Security-Safety Arguments

- **Unstated assumptions:** "Displayed vitals are accurate" assumes the communication channel is trustworthy, but this is never stated or defended
- **Circular reasoning:** "The system is secure because it passed the security test", but the test scope may not cover the relevant attack paths
- **Confusing compliance with assurance:** ISO 27001 certified does not mean safety-relevant threats have been addressed
- **Ignoring undeveloped nodes:** marking a sub-goal as undeveloped and forgetting it leaves a gap in the safety case
- **Stale evidence:** a penetration test from two years ago does not demonstrate current security posture

Challenges of Security-Informed Safety

- **Different expertise and culture:** safety engineers think in failure modes, security engineers in attack trees and adversary models. Few are fluent in both
- **Standards lag behind practice:** IEC 61508 predates connectivity, ISO 27001 ignores physical safety consequences, IEC 62443 is newer but not yet universal
- **Security evolves, safety cases are static:** a safety case is validated at a point in time, but new vulnerabilities emerge daily. Patching may invalidate certification
- **Cost and legacy systems:** retrofitting security into deployed safety-critical systems is expensive and risky. OT lifecycles are 15–30 years

Regulatory Drivers

- **EU Cyber Resilience Act (2024):** mandatory cybersecurity requirements for products with digital elements, including IoT devices
- **UK PSTI Act (2024):** bans default passwords and requires vulnerability disclosure for consumer IoT
- **FDA Pre-market Guidance (2023):** medical device manufacturers must submit cybersecurity documentation as part of safety approval
- **IEC 62443:** industrial automation security standard that integrates with IEC 61508 safety requirements
- **UNECE WP.29 (2021):** UN regulation requiring cybersecurity management systems for type approval of road vehicles

Summary

- **Security-Informed Safety:** assurance is incomplete without accounting for security threats
- **Attacks $\neq$ failures:** adversaries violate the independence and randomness assumptions of traditional safety analysis
- **Causal chain:** vulnerability $\rightarrow$ exploit $\rightarrow$ compromise $\rightarrow$ hazard $\rightarrow$ harm
- **Safety cases** use **CAE** to build structured assurance arguments
- **GSN** provides a standard graphical notation for safety case arguments
- **Security-informed hazard analysis** extends traditional methods by treating attack paths as hazard sources
- **Regulation is catching up:** EU CRA, UK PSTI, FDA guidance, UNECE WP.29 all require integrated security-safety thinking

Key references

- Bloomfield, R., Netkachova, K., & Stroud, R. (2013). Security-Informed Safety: If It's Not Secure, It's Not Safe. SAFECOMP 2013. **[Link]**
- Halperin, D. et al. (2008). Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. IEEE S&P. **(Link)**
- Kelly, T. (1998). Arguing Safety: A Systematic Approach to Managing Safety Cases. PhD thesis, University of York. **[Link]**
- IEC 61508. Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems. **[Link]**