Resilience and Mitigation

Dr Ayman El Hajjar

CyBOK Mapping

The lecture maps to the following CyBOK Knowledge Areas:

Full Coverage:

- **Infrastructure Security → Cyber-Physical Systems Security**
- **Attacks and Defences → Security Operations & Incident Management**
- **Human, Organisational and Regulatory Aspects → Risk Management & Governance**

Partly Covered:

- **Infrastructure Security → Network Security**
- **Infrastructure Security → Physical Layer and Telecommunications Security**
- **Systems Security → Authentication, Authorisation & Accountability**

Outline

Resilience in Safety-Informed Security

What is Resilience?

Resilience

Resilience is the ability of a system to continue providing an acceptable level of service when faults, attacks, disruption, or uncertainty affect normal operation.

- In safety-informed security, resilience is not only about keeping the system running.
- The system must remain **safe**, **constrained**, **observable**, and **recoverable**.
- Security controls may fail, but the system should still avoid unsafe behaviour.
- Resilience therefore connects security controls to safety consequences.

Resilience assumes that prevention may be incomplete and asks what the system does next.

From Prevention to Resilience

Prevention Focus

- Stop unauthorised access
- Block malicious traffic
- Prevent unsafe commands
- Harden devices and networks

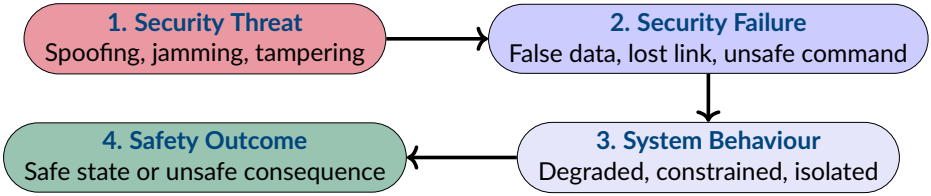
Resilience Focus

- Detect abnormal behaviour
- Limit the effect of compromise
- Maintain safe degraded operation
- Recover with evidence and assurance

- Prevention reduces the likelihood of security failure.
- Resilience reduces the **safety impact** when failure still occurs.
- Safety-critical systems need both.

The question is not only whether an attack can be blocked, but whether the system remains safe if it is not.

Security Failure and Safety Impact



- A security failure does not automatically become a safety incident.
- The system architecture determines how failure is **contained**, **translated**, or **amplified**.
- Resilience mechanisms aim to keep the path from security failure to safety harm under control.

When a wireless or IoT component is compromised, what prevents the system from moving into an unsafe state?

Designing for Degraded Operation

Why Degraded Operation Matters

Degraded Operation

A controlled mode in which a system continues to provide limited functionality when normal operation is unsafe, unavailable, or no longer trustworthy.

- Wireless and IoT systems often depend on communication links, gateways, sensors, cloud services, and remote commands.
- Dependencies can fail through faults, interference, misconfiguration, or attack.
- In safety-critical contexts, the system should not simply stop, continue blindly, or accept untrusted inputs.
- It should move into a **defined degraded mode** with clear operational limits.

Degraded operation is safer when it is designed, documented, tested and understood before an incident occurs.

Graceful Degradation

Graceful Degradation

A system reduces capability in a controlled way rather than failing abruptly or continuing unsafe operation.

- The system keeps essential safety functions available where possible.
- Non-essential functions may be disabled, delayed, or moved to manual control.
- The degraded state should be visible to operators and maintainers.
- Recovery should require checks before returning to normal operation.

Example

A medical monitoring device loses its wireless gateway. It stores readings locally, alerts staff and limits remote-control functionality until the trusted connection is restored.

Fail-Safe, Fail-Secure and Fail-Operational

Fail-Safe
Moves to a state that reduces risk to people, equipment, or the environment.

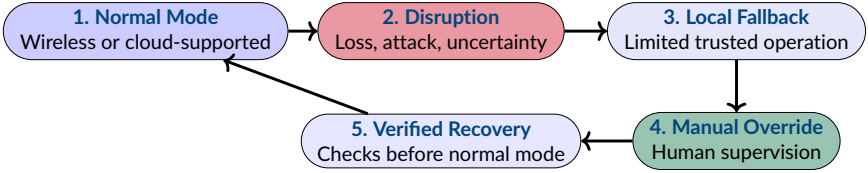
Fail-Secure
Preserves security properties such as access control, confidentiality, or integrity.

Fail-Operational
Continues operating, possibly with reduced function, because stopping may be unsafe.

- These goals can conflict in safety-critical systems.
- Locking a device may be **secure**, but not always **safe**.
- Continuing operation may preserve service, but may amplify harm if inputs are compromised.
- The correct behaviour depends on the hazard, context, timing and available fallback options.

When a security control fails, should the system stop, restrict operation, continue locally, or request human intervention?

Local Fallback and Manual Override



- Local fallback reduces dependence on remote services during disruption.
- Manual override can be essential when automated decisions are no longer trustworthy.
- The override itself must be controlled, authenticated and logged.

Fallback is not a bypass. It is a controlled operating mode with explicit authority, limits and evidence.

Mitigation Patterns for Wireless and IoT Systems

Defence in Depth

Defence in Depth

Uses **multiple independent controls** so that failure of one control does not directly expose the system to unsafe behaviour.

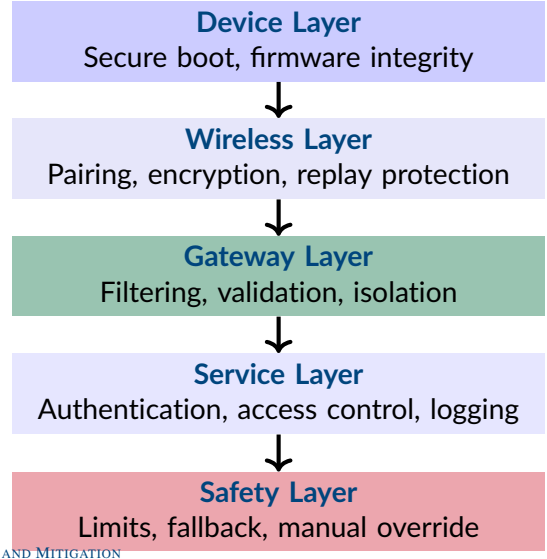
- Wireless and IoT systems should not rely on one protective boundary.
- Devices, gateways, networks, applications and data flows all need controls.
- Independent controls limit how far one compromise can spread.
- Each layer should reduce the chance of a **safety-relevant failure**.

Defence in depth is not just more controls. It is layered control of how compromise can propagate.

Layered Controls in an IoT System

- Technical controls reduce exposure at each layer.
- Safety controls constrain what can happen when security controls fail.
- The safety layer should not rely blindly on the same compromised data or communication path.

Safety-informed view: the system should not allow a compromised wireless message to become an unsafe physical action without further checks.



Segmentation and Isolation

Segmentation

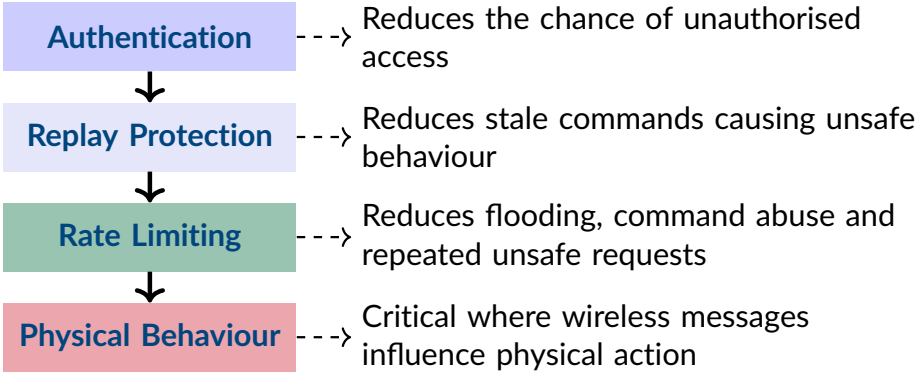
Divides the system into controlled zones so that compromise in one area does not automatically spread to others.

- Critical devices should not share unrestricted networks with general systems.
- Gateways can mediate traffic between devices, clinical systems, cloud services and maintenance interfaces.
- Isolation limits the **blast radius** of compromised devices or credentials.
- Access should depend on identity, role and device context, not location alone.

Example

A hospital separates patient monitors from guest WiFi. The gateway accepts authenticated device traffic and logs clinical access.

Authentication, Replay Protection and Rate Limiting



Can a valid-looking message still be unsafe because it is old, repeated, excessive, or outside the expected context?

Secure Updates and Configuration Control

Secure Update

A **secure update** mechanism ensures that only authorised, authentic and integrity-protected software or configuration changes are installed.

- Wireless and IoT systems often need **long operational lifetimes**.
- **Vulnerabilities** may be discovered after deployment.
- **Updates** can mitigate risk, but update mechanisms can also become **attack paths**.
- **Configuration changes** should be authenticated, logged and reversible where possible.

Safety-informed view: a patch should not only be secure to install. It must also preserve the safety assumptions used in the assurance argument.

Redundancy, Diversity and Fallback

Redundancy and Diversity

Redundancy provides alternative components, channels, or paths for a function. **Diversity** reduces shared failure modes.

- **Redundancy** can help maintain service when a device, sensor, gateway, or communication path fails.
- **Diversity** is needed when the same vulnerability, configuration error, or attack can affect all redundant elements.
- In wireless and IoT systems, redundancy may include **backup links**, **alternative gateways**, local control, or secondary sensors.
- Redundancy should support **safe continuity**, not uncontrolled continuation.

Redundancy improves resilience only when it avoids **common-mode failure** and supports a **defined fallback behaviour**.

Voting and Plausibility Checking

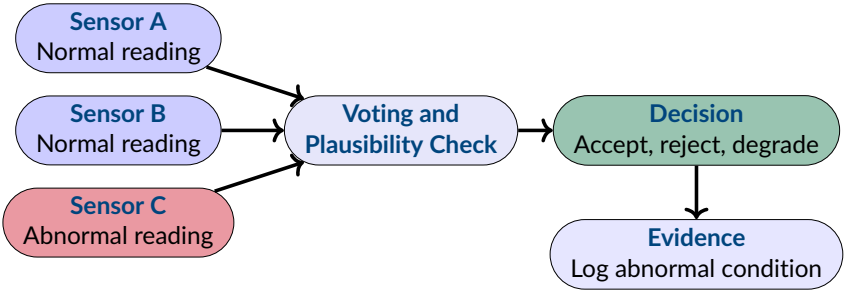
Plausibility Checking

Voting compares multiple sources before accepting a value. **Plausibility checking** compares values or commands against expected physical, operational, or contextual constraints.

Voting	Plausibility
Uses multiple independent sources to accept, reject, or weight a reading.	Checks values, rates of change and commands against expected context.

Safety-informed view: a valid sensor value may still be physically implausible or unsafe to use.

Redundant Sensor Decision



- The system should not rely on a **single sensor reading** when it controls safety-relevant behaviour.
- **Disagreement** between sources may trigger degraded operation, manual review, or temporary isolation.
- **Logs** provide evidence for diagnosis, recovery and assurance review.

Common-Mode Failure

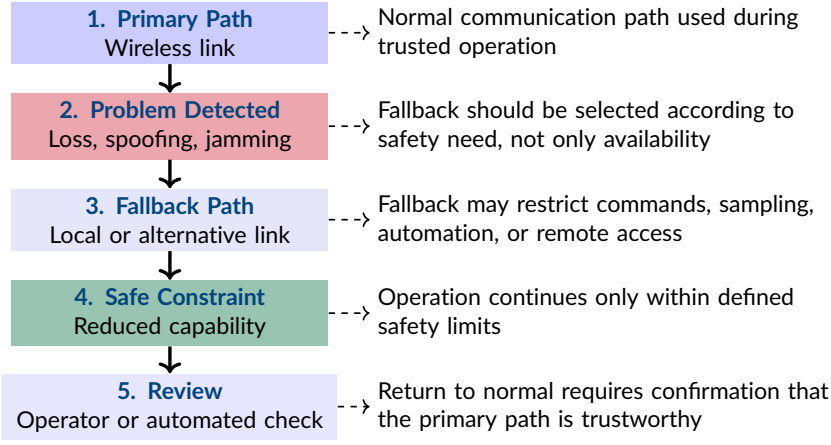
Common-Mode Failure

Occurs when multiple components fail through a **shared cause**, such as a dependency, vulnerability, assumption, or condition.

- Sensors may fail together if they share **firmware**, calibration, or wireless protocols.
- Communication paths may fail together if they share a **gateway**, power supply, or cloud service.
- Software implementations may fail together if they share a **vulnerable library**.
- Attackers may target **shared dependencies** to defeat apparent redundancy.

Are the redundant elements genuinely independent, or do they fail together under the same attack or operating condition?

Communication and Sensor Fallback



Fallback should preserve **safety assumptions**, not simply reconnect through any **available path**.

Monitoring, Detection and Response

Monitoring

Analyses communication, device behaviour and system state to detect **abnormal** or **unsafe conditions**.

- Wireless and IoT monitoring is difficult because devices may be **constrained**, mobile, intermittent, or exposed.
- Evidence may come from gateways, radio observations, logs and operator reports.
- Monitoring should cover **security indicators** and **safety-relevant behaviour**.
- Detection is useful only when it triggers a **defined response**.

Monitoring should make degradation and compromise visible before they become unsafe physical behaviour.

Wireless and IoT Indicators

Wireless Indicators

- Signal loss or interference
- Association failures
- Unusual retransmissions
- Spoofing or replay signs

Device Indicators

- Unexpected traffic volume
- Changed command patterns
- Abnormal sensor readings
- Firmware or resource anomalies

Safety-informed view: a weak signal, abnormal sensor value, or unusual traffic pattern matters when it affects trust in system behaviour.

Anomaly Detection

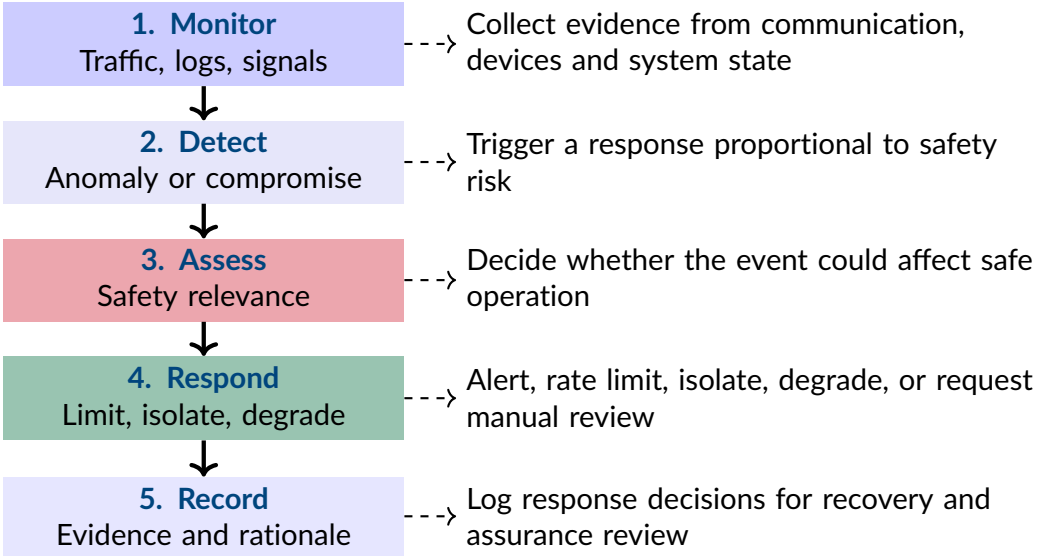
Anomaly Detection

Identifies behaviour that deviates from **expected communication**, device, user, or physical patterns.

- **Rule-based detection** checks known thresholds, sequences, or safety limits.
- **Signature-based detection** identifies known attacks or malicious patterns.
- **Behavioural detection** compares current activity with expected system behaviour.
- In safety-critical systems, **false positives** and **false negatives** both affect operation.

What should the system do when it is uncertain whether behaviour is malicious, faulty, or simply unusual?

From Detection to Safe Response



Incident Response

The planned process for **detecting**, **containing**, investigating, **recovering from** and learning from a security incident.

- **Detection:** Identify alerts, abnormal behaviour, or safety-relevant deviations.
- **Containment:** Isolate affected devices, networks, credentials, or services.
- **Evidence Preservation:** Preserve logs, captures, configurations and firmware records.
- **Recovery:** Restore trusted operation after **verification**.
- **Review:** Update controls, procedures, assumptions and **assurance evidence**.

In safety-critical IoT, incident response must **protect people** and **preserve evidence** at the same time.

Recovery, Assurance and Residual Risk

Recovery and Restoration

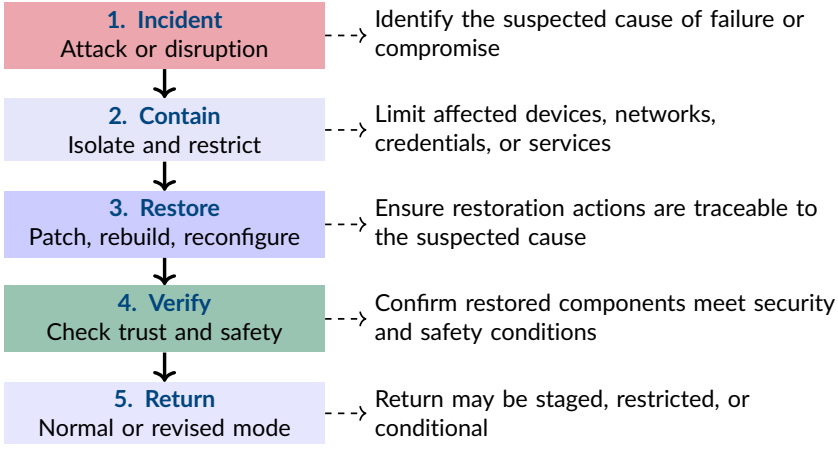
Recovery

The process of restoring a system to a **trusted** and **acceptable operating state** after disruption, compromise, or degraded operation.

- Recovery is not simply **restarting** the system or reconnecting a device.
- The system must be **checked** before it returns to normal operation.
- Recovery may require **firmware validation**, configuration review, credential replacement, log analysis and operator approval.
- In safety-critical contexts, recovery must restore both **security confidence** and **safety confidence**.

A system should not return from degraded mode to normal mode until relevant **trust assumptions** have been re-established.

Verified Return to Operation



Safety-informed view: recovery is complete only when the system can be trusted to operate within its safety assumptions again.

Assurance Impact After an Incident

Assurance Impact

Occurs when an incident changes confidence in **claims**, assumptions, evidence, or controls used to justify system safety and security.

- Incidents may reveal that a **threat** was underestimated.
- Evidence may no longer support the original **claim** if the system has changed.
- A mitigation may work technically but still leave an unacceptable **operational risk**.
- The **assurance case** should be updated when assumptions, controls, or residual risks change.

An incident is not only operational. It may invalidate part of the **security** or **safety argument**.

Residual Risk

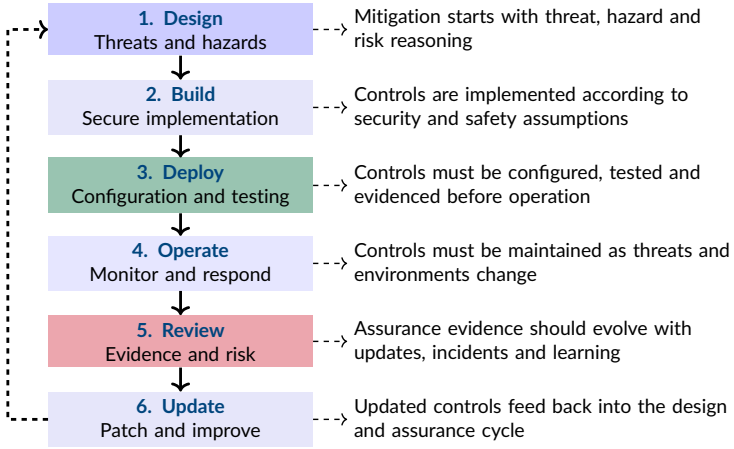
Residual Risk

The risk that remains after **controls**, mitigations and recovery measures have been applied.

- No mitigation removes **all risk**.
- Residual risk should be **explicit**, justified and accepted at the appropriate level.
- Safety-critical systems require evidence that remaining risk is **tolerable in context**.
- Residual risk may change when threats evolve, systems are updated, or deployment conditions change.

What risk remains after mitigation, and who has **authority** to accept it?

Mitigation Across the System Lifecycle



Resilience depends on maintaining controls, evidence and assumptions throughout the system lifecycle.

Summary

- **Resilience** maintains acceptable, safe and recoverable operation during faults, attacks, disruption, or uncertainty.
- **Degraded operation** should be designed in advance, with clear limits, fallback modes and recovery conditions.
- **Defence in depth** reduces the chance that one compromised component, message, or control causes unsafe behaviour.
- **Redundancy and diversity** support continuity when common-mode failure is understood and controlled.
- **Monitoring and detection** connect security indicators to safety-relevant system behaviour.
- **Incident response and recovery** protect people, preserve evidence and restore trust before normal operation resumes.
- **Residual risk** remains after mitigation and must be justified, evidenced and accepted.

Key References

- NIST. (2021). *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach*. NIST SP 800-160 Vol. 2 Rev. 1. **[Link]**
- NIST. (2024). *Cybersecurity Framework (CSF) 2.0*. **[Link]**
- NCSC. *Cyber Assessment Framework (CAF)*. **[Link]**
- NCSC. *Operational Technology Guidance*. **[Link]**
- ENISA. (2019). *Good Practices for Security of IoT: Secure Software Development Lifecycle*. **[Link]**
- IEC 62443. *Security for Industrial Automation and Control Systems*. **[Link]**
- FDA. *Cybersecurity in Medical Devices*. **[Link]**
- MITRE. *ATT&CK for ICS*. **[Link]**