Information Security Concerns in Aircraft

AFRL Safe & Secure Systems & Software Symposium (S5)
June 3, 2009

Darren Cofer

Rockwell Collins

Outline

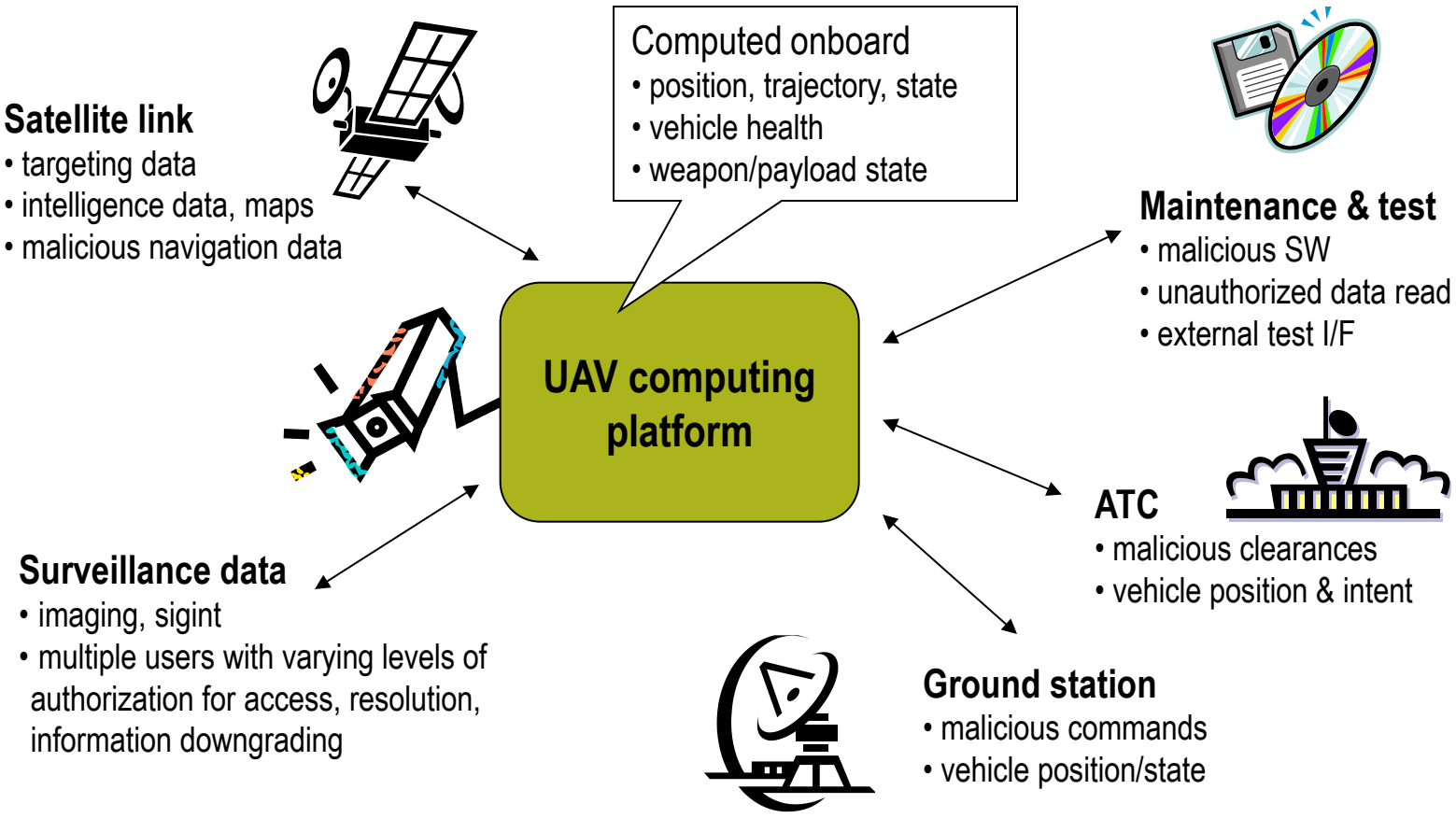
- Security concerns in aircraft
  - It's about safety and information integrity, not confidentiality
- SC-216
  - Security concerns for commercial aircraft
  - Model and requirements
- Mitigation
  - Cross-domain guard
  - Information flow analysis

Overview

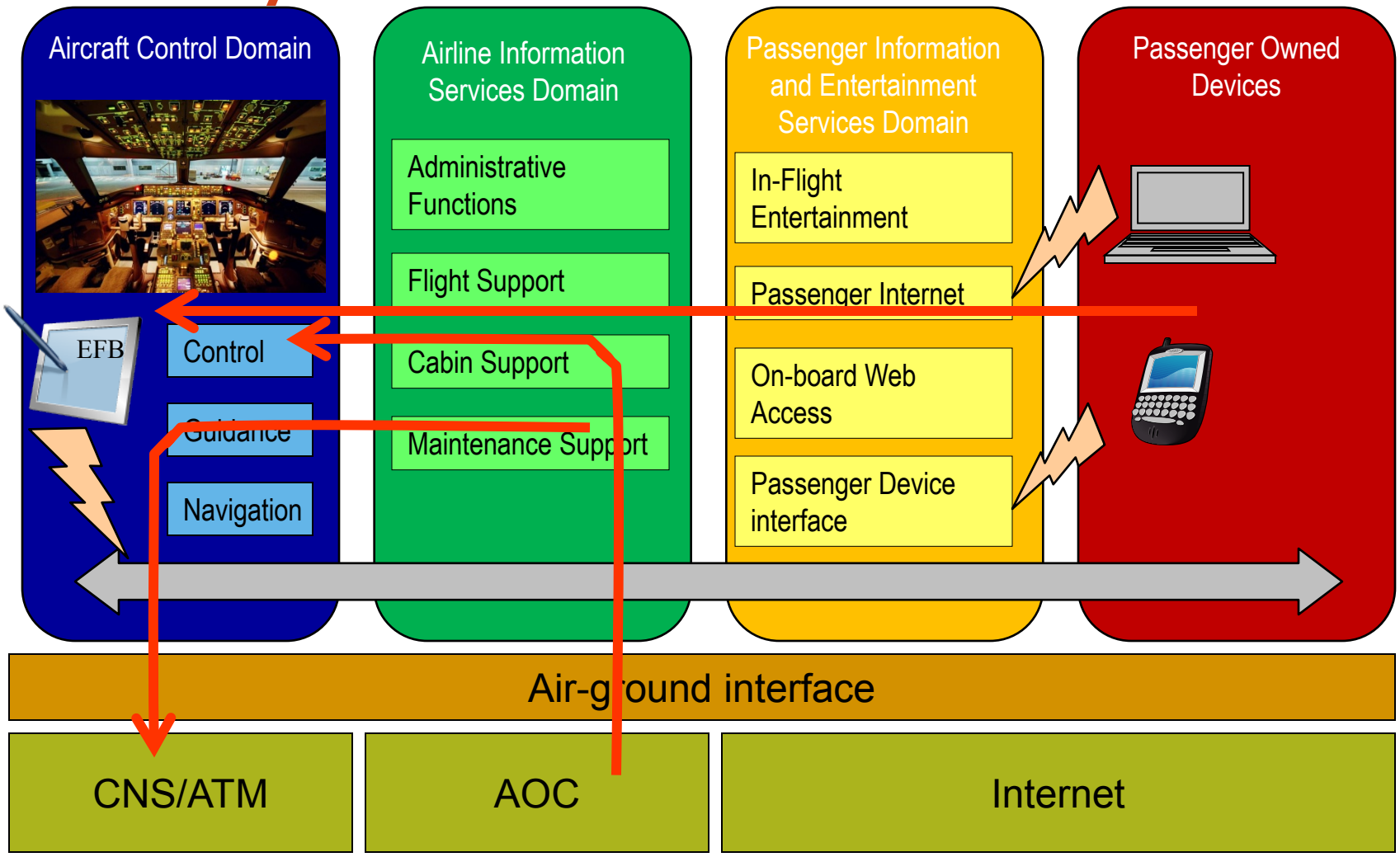
- Security concerns in our context are not *primarily* related to protection of information (confidentiality).
- The focus is on threats to the vehicle (safety of flight) that may arise from security-related threats and vulnerabilities.
- In many cases, the architectural mechanisms and analysis methods that guarantee information integrity will also support confidentiality.
  - encryption, authentication, separation/non-interference…

Security concerns for UAVs

- Data flows in/out of vehicle may represent security vulnerabilities
- Many of these can impact vehicle safety
- Similar for manned military aircraft



Security concerns for commercial aircraft

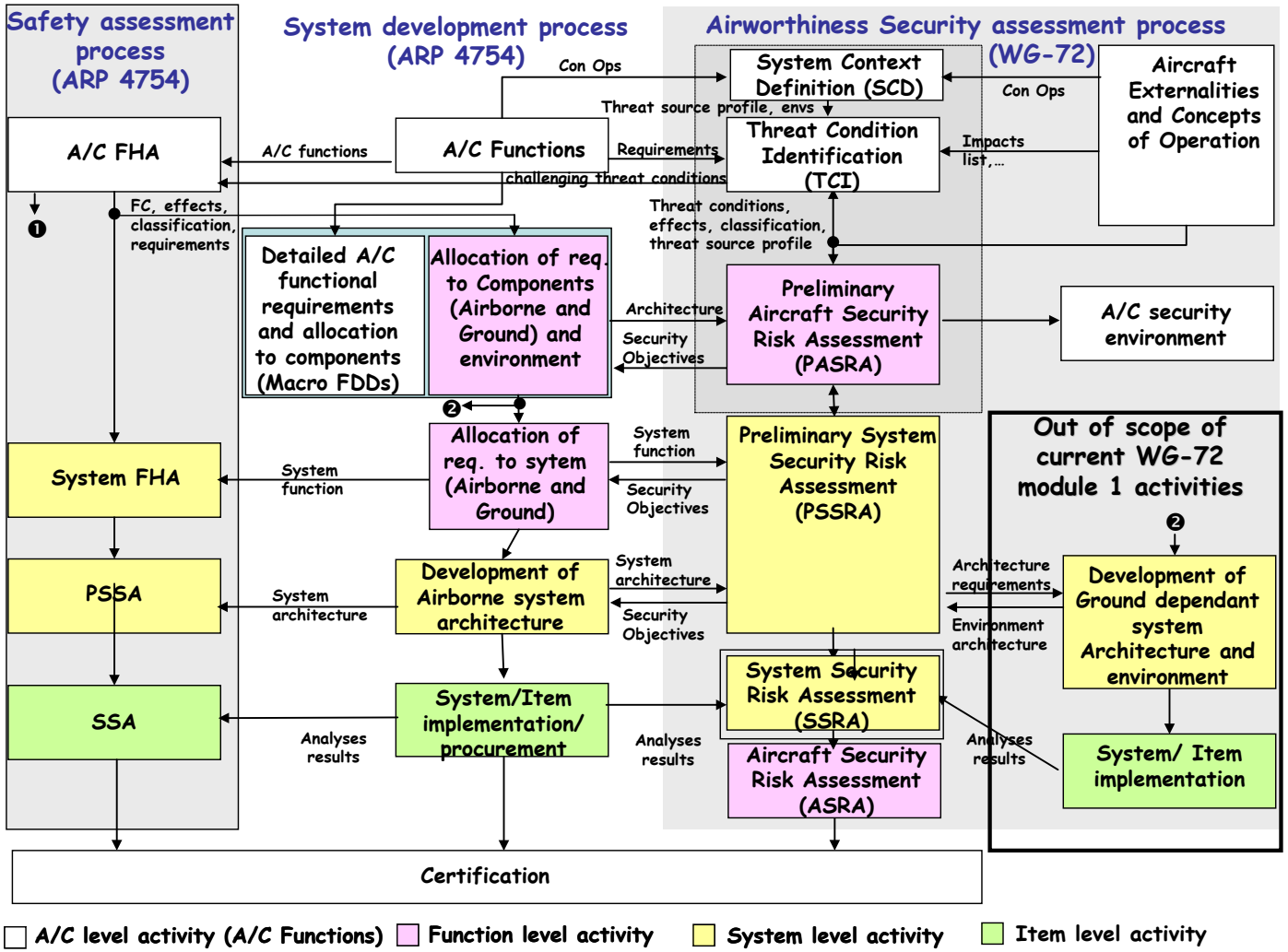


Rockwell Collins

SC-216

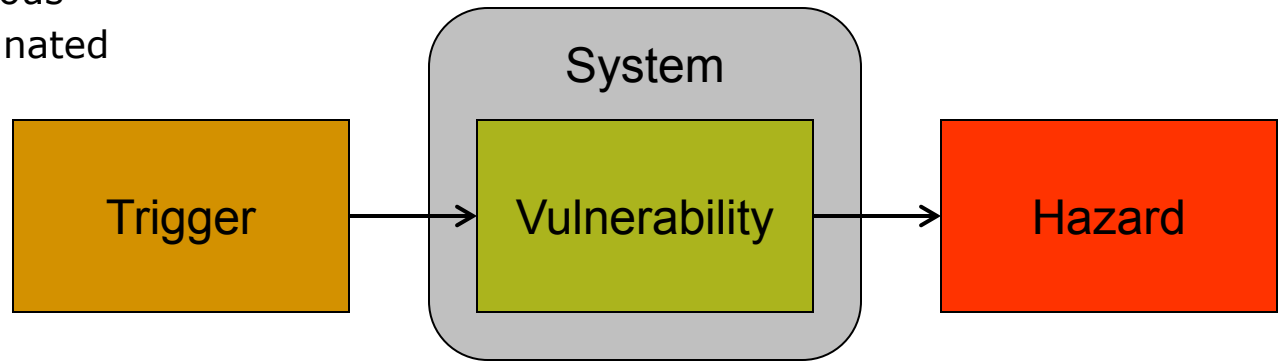
- RTCA committee developing guidance for security requirements and assessment processes for commercial aircraft
  - joint activity with Europe
- Address integration with safety assessment processes
- Draft documents available for review

Airworthiness Security Assessment Process



Assessment of safety/security risks

- Safety and security analysis are similar
  - Both seek to identify vulnerabilities/flaw in the system design that may be triggered by threats/failures
- But there are important differences
  - In safety analysis, triggering conditions (inputs or HW faults) are treated as
    - random
    - unintentional
    - independent (unless common cause is identified)
  - In security analysis, threats must be assumed to be
    - deliberate
    - malicious
    - coordinated



Some characteristic of security hazards

- A function may have a vulnerability that can be assumed to be error-free under for safety analysis, but not threat-free
- Some failure conditions may be excluded in safety analysis due to low probability, but not in security analysis
  - e.g., erroneous packet that matches CRC
- Security is an emergent property in a complex system
  - vulnerability in one system may be exploited to launch a later attack against another system
- Security assessment must consider external attackers and those elements of system which are exposed
  - cannot limit scope to onboard systems: ground systems important
- Must consider vulnerabilities throughout lifecycle, not just development
  - operational and management practices are important

Threats and countermeasures

- Denial of service
  - overwhelm communications or processing with malicious requests
  - strong partitioning of resources
  - redundancy may be used to make attacks more difficult
  - fiber optic network with wavelength multiplexing
    - acts like physically separate network
- Malware
  - installed during development or maintenance
  - certificates to validate new code
  - 'virus scanning' mechanisms
    - extra processing time
- Message spoofing
  - malicious commands or data
  - encryption/authentication
    - size/weight/power are concerns, especially for small UAVs

Cross domain solutions

- Permit communication between security domains
  - but in a controlled and secure way
- High assurance guard
  - run-time enforcement of security policy
- Information flow analysis
  - design-time detection/elimination of vulnerabilities

Cross-domain guard

- A guard mediates information flow between different security domains according to a specified policy

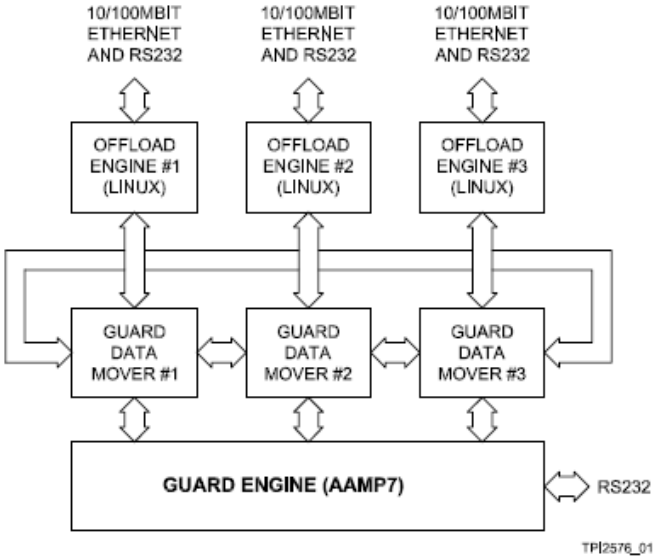
one-way flow
("diode")

filtered flow
(data classification)

transformed flow
(downgrading/scrubbing)

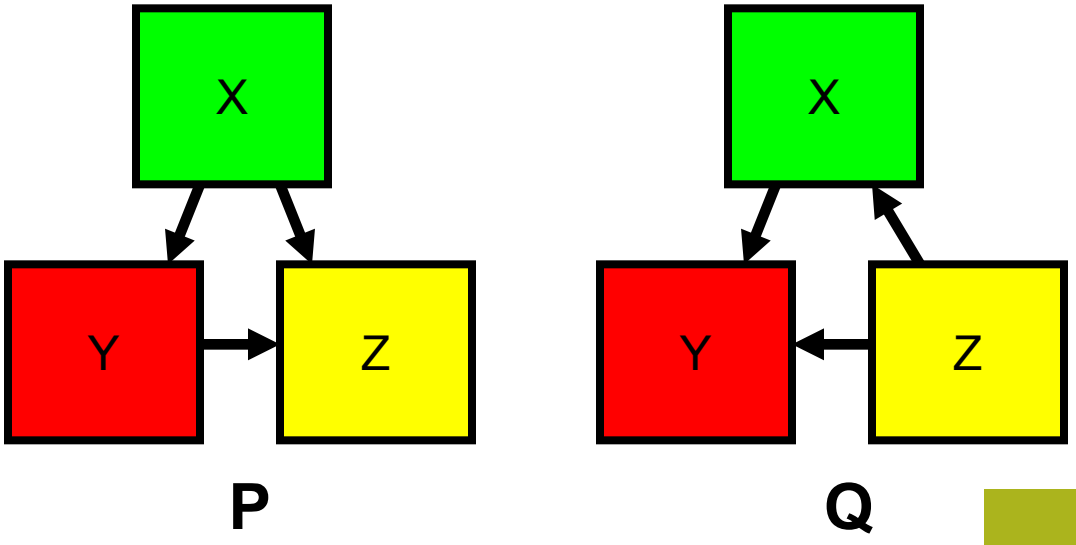
Example: Turnstile high-assurance guard

- Provably correct information flow (and other behavioral properties)
- Minimizes size of core guarding functionality that must be analyzed
  - Guard Engine (GE)
  - GDMs slaved to GE
  - Precise control over what data flows where and under what conditions



Example: Conditional Information Flow

In a multi-threaded system, two independent entities ("principals" P and Q) are able to move information between storage devices X,Y, and Z as described by the arrows below.



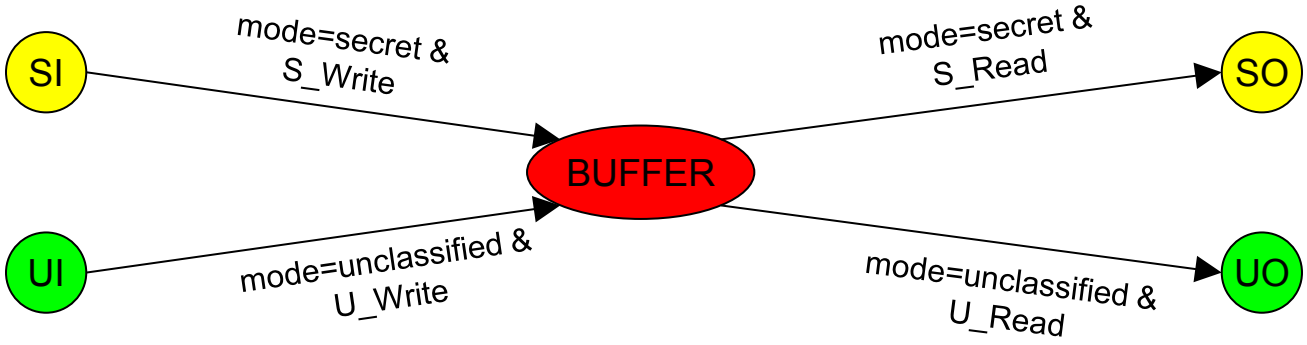
Q: Can information from Y reach X?

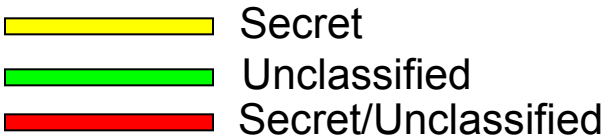
A: It depends…

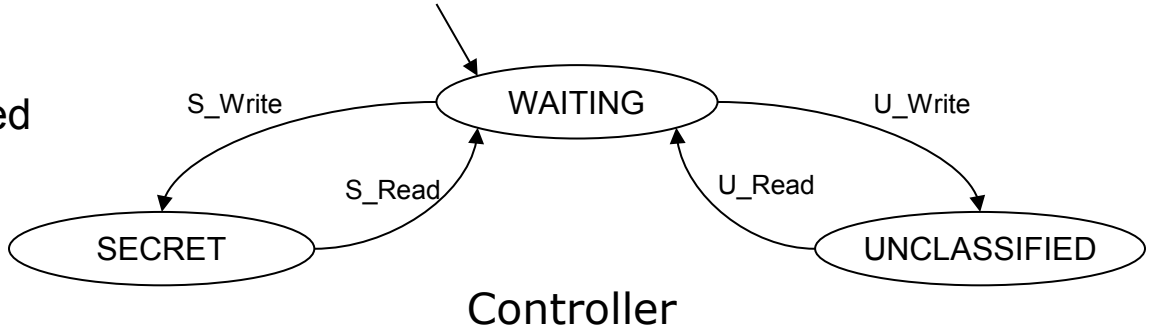
GOAL:
Find/eliminate
vulnerabilities

Modeling Conditional Information Flow

- Both a secret and unclassified entity can read and write to a shared buffer
- This buffer must be safeguarded by a controller to prevent undesirable information flow

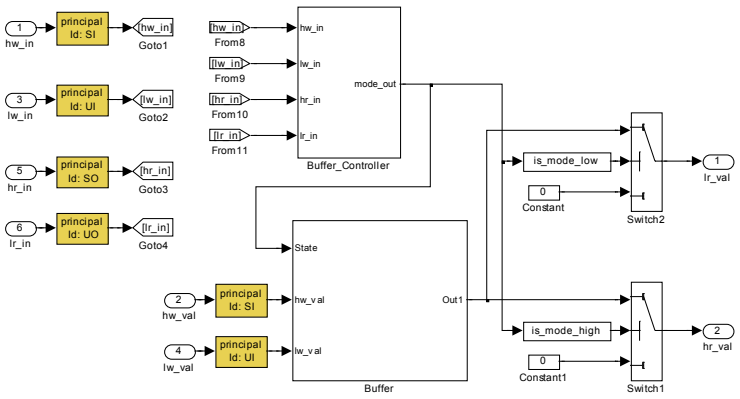


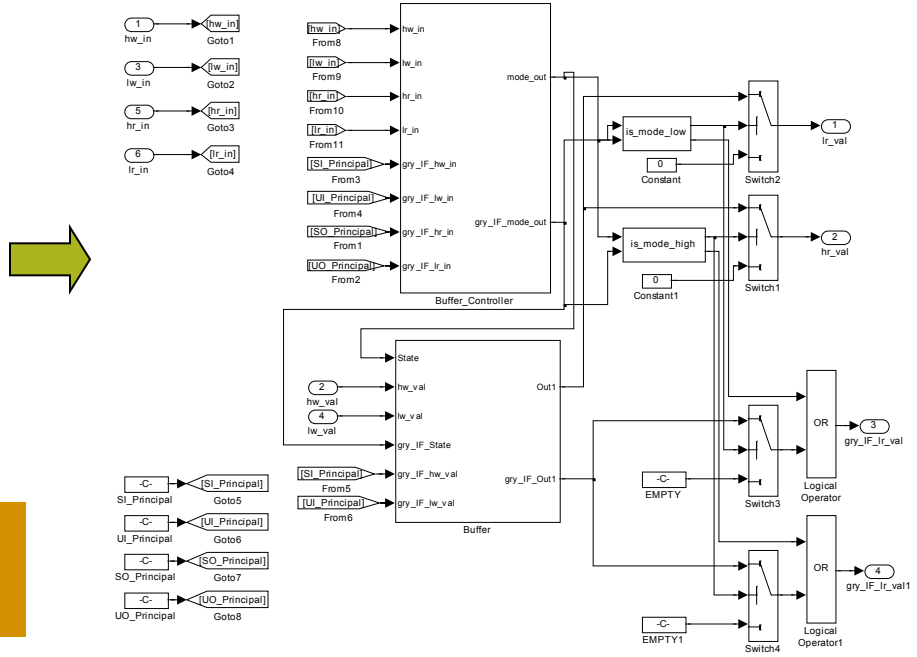




Generating Information Flow Models

- Creating information flow models by hand
  - …is time consuming
  - …is error prone
  - Each new property may require a different model
- Gryphon-IF tool: automatically generates information flow models from Simulink functional models



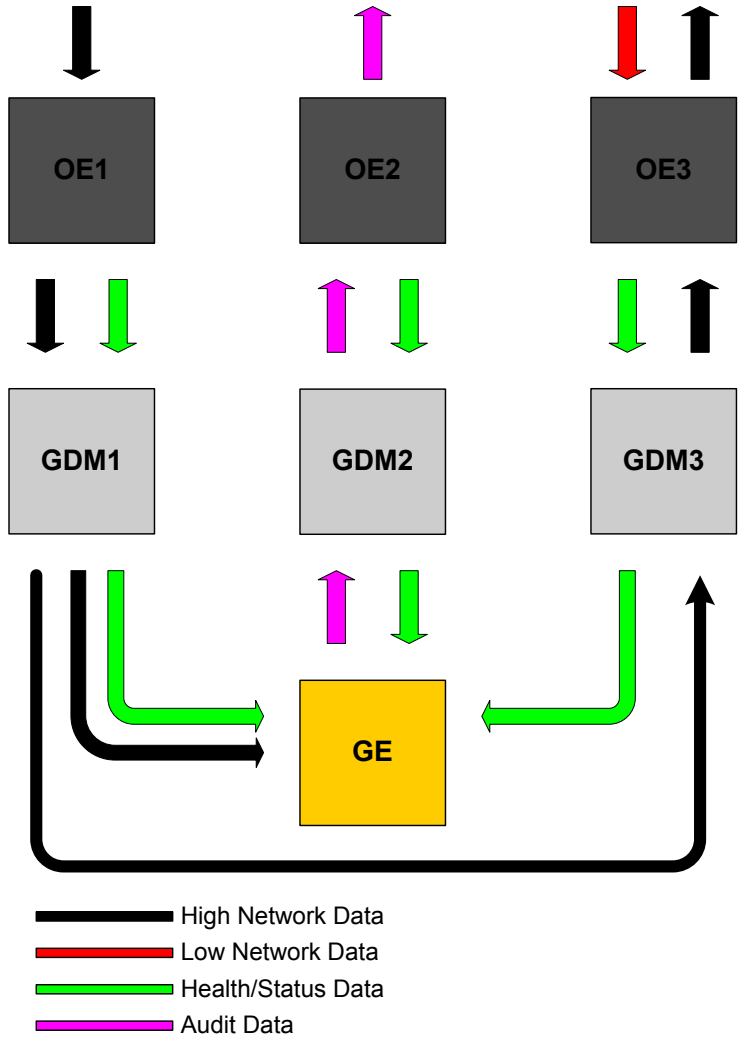


Analysis checks information flow from principal signals

Information flow analysis in Turnstile

- Modeled and analyzed part of the Turnstile architecture in Simulink
  - High, Low, and Audit GDMs, OEs
  - Guard Engine

- Properties Checked
  - Information from the Low (OE3) network cannot be perceived by the High (OE1) network.
    - not gry_IF_OE1_TX_Access[p_oe3_writer] ;
    - not gry_IF_OE1_TX_Access[p_oe3_reader] ;
    - not gry_IF_OE1_RX_Access[p_oe3_writer] ;
    - not gry_IF_OE1_RX_Access[p_oe3_reader] ;
    - …
  - Information from the Low network cannot be perceived by the Audit (OE2) network.
    - not gry_IF_OE2_Audit_Access[p_oe3_writer] ;
    - not gry_IF_OE2_Audit_Access[p_oe3_reader];
    - not gry_IF_OE2_Audit_Data[p_oe3_writer] ;
    - not gry_IF_OE2_Audit_Data[p_oe3_reader] ;
    - not gry_IF_OE2_CTRL_Access[p_oe3_writer]
    - …
  - Checked 22 properties
    - One violation: The 'high' TX_Access variable can perceive the 'low' OE_reader through a ready bit
    - This is a known channel
      - Supports Quality of Service



Key concepts and benefits

- "Push Button" information flow analysis for Simulink Models
  - Analysis is completely automated through Gryphon translator and model checker
  - Able to model multi-threaded, distributed architectures
  - Can detect:
    - Timing channels
    - Unintended direct information flows
- Precise
  - Determine dependencies of individual elements of aggregate data structures
  - Considers conditional information flow
- Sound
  - Conservative abstraction of actual information flow in model
- Practical
  - Supports most Simulink/Stateflow blocks
  - Supports several different model-checking and theorem proving back-end tools
  - Existing Gryphon optimizations allow efficient analysis of large models
  - Turnstile architecture model checks in a few seconds with Prover

Conclusions

- Security vulnerabilities impact aircraft safety
- Concerns are similar for commercial and military aircraft
  - excluding confidentiality concerns
- Safety and security threat assessments are related
  - but have different underlying assumptions
- Guidance is being developed for integration into commercial aircraft certification process
- Formal analysis of software and system designs can contribute to security assessment and elimination of vulnerabilities