Security Assurance Cases

SCC @ HCSS 2011

Overview

- CESG the organization
- Cryptographic product evaluation and certification
- Software Assurance Capability Development Project (formerly called High Assurance Software Research)
- The report "*Security Assurance Cases: Motivation and the State of the Art*" we commissioned from York University (UK)

CESG

- CESG: UK Government's 'National Technical Authority' for IA
  - Wide range of activities and services
    - Publish policy and guidance documents
    - Technical advice to Govt departments and Industry
    - Emergency response team for security incidents
    - Penetration testing service
    - Assurance services, including product evaluation and certification, partnering with UK Industry
    - www.cesg.gov.uk

Repayment model

- Repayment model
  - Since 1997
  - charge for services but non-profit, so ‘half’ a business
  - Impact on research
  - Impact on disseminating advice to wider areas of UK government, beyond our ‘traditional’ customers
  - Now (partly) moving away from this but still intend to charge for product evaluation/certification services

CAPS Evaluation scheme

- CESG Assisted Products Scheme (CAPS)
  - Evaluation and certification service primarily for UK vendors
  - Approves cryptographic products for use by UK Government
  - Includes offering advice and guidance to developers
  - Evaluation process generally places considerable emphasis on manual code-inspection
  - Vendors are requested to fix issues/problems
  - Advisory powers only, no legal regulatory framework
  - Vendor incentives to create sufficiently secure products
    - Getting the ‘badge’, can help with exports
    - Evaluation cost/time

Typical products

- IP (network) cryptos
- File/Disk encryptors for data at rest
- Secure mobile comms devices

Technical Issues

- Size, Complexity and richness of features
- Increased use of software and FPGAs.
- Post-development security analysis vs Building Security In
- Enormous variation in degree of engagement with developers
- Trend towards claims, arguments evidence approach?

Where do I work?

- Software Assurance Capability Development (SACD) project
  - Small research area
- Vision for team
  - Investigate relevant techniques, both current and new that can make a positive difference to the production of software and the evaluation of software for crypt related products.
  - Work to promote best practice in IA software engineering techniques across academia, industry and government.
- Try to expose our evaluators to ‘new’ ideas, tools and techniques, but ultimately their decision regarding adoption.

Current work

- Formal methods
  - applying ALLOY tool (from MIT) to system-level design
  - Model abstractions of systems and checks asserted properties, providing counter-examples if they do not hold.
- Software engineering standards and processes
  - Cross-over between safety and security domains
  - Claims, Arguments, Evidence approach to certification
  - York University work on security assurance cases
    - Applicability of assurance cases to security domain
    - Use of GSN to present structured arguments

Assurance Cases

- Widely used in the safety domain for 30 years
- Justify the safety of a system and render that justification open to review
- Provide a record of why we believed at evaluation time that the system was adequately safe
- If produced early in the development lifecycle, may aid in producing an ‘evaluatable’ design and implementation.
- Not been used much in the security domain.

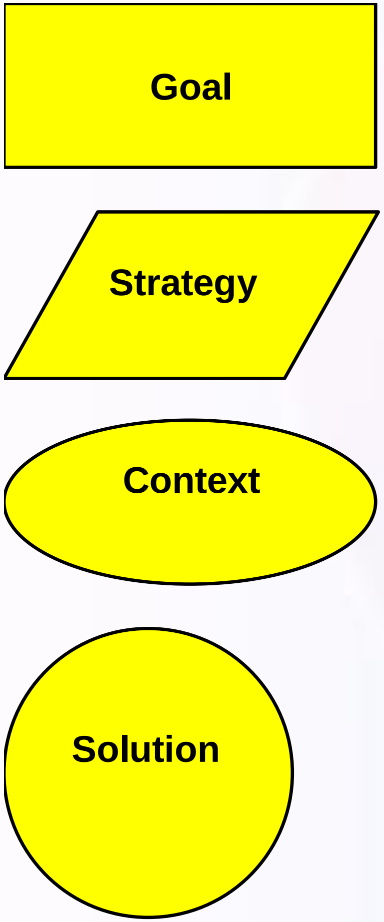
What is an assurance case?

- A structured argument showing how a high-level claim is supported by detailed evidence of low-level properties
- Shows how different types of evidence combine (eg analysis, testing, design review, process)
- Makes the subjective reasoning or rationale, that connects evidence and claims, explicit.
- Capture assumptions
- Requires critical thinking, should not become a tick-box exercise
- Does not replace any specific technique for analysis or generating evidence. It shows the connection between those techniques and the high-level claims we want to make.

Notations for assurance cases

- Can use natural language, but it is easy to find examples in which it is quite difficult to follow the chain of reasoning (ambiguity, not always concise, hard to perceive overall structure).
- Structured argument notation
  - Goal structuring notation (GSN). Forthcoming GSN standard
  - Claims, Arguments, Evidence (CAE from Adelard)
  - Clearly shows chains of reasoning and dependencies between claims
  - Improves comprehension amongst developers, independent assessors and certification authorities

Principal GSN elements



Goal

States a claim

Strategy

Nature of argument that connects goals to sub-goals (reasoning step)

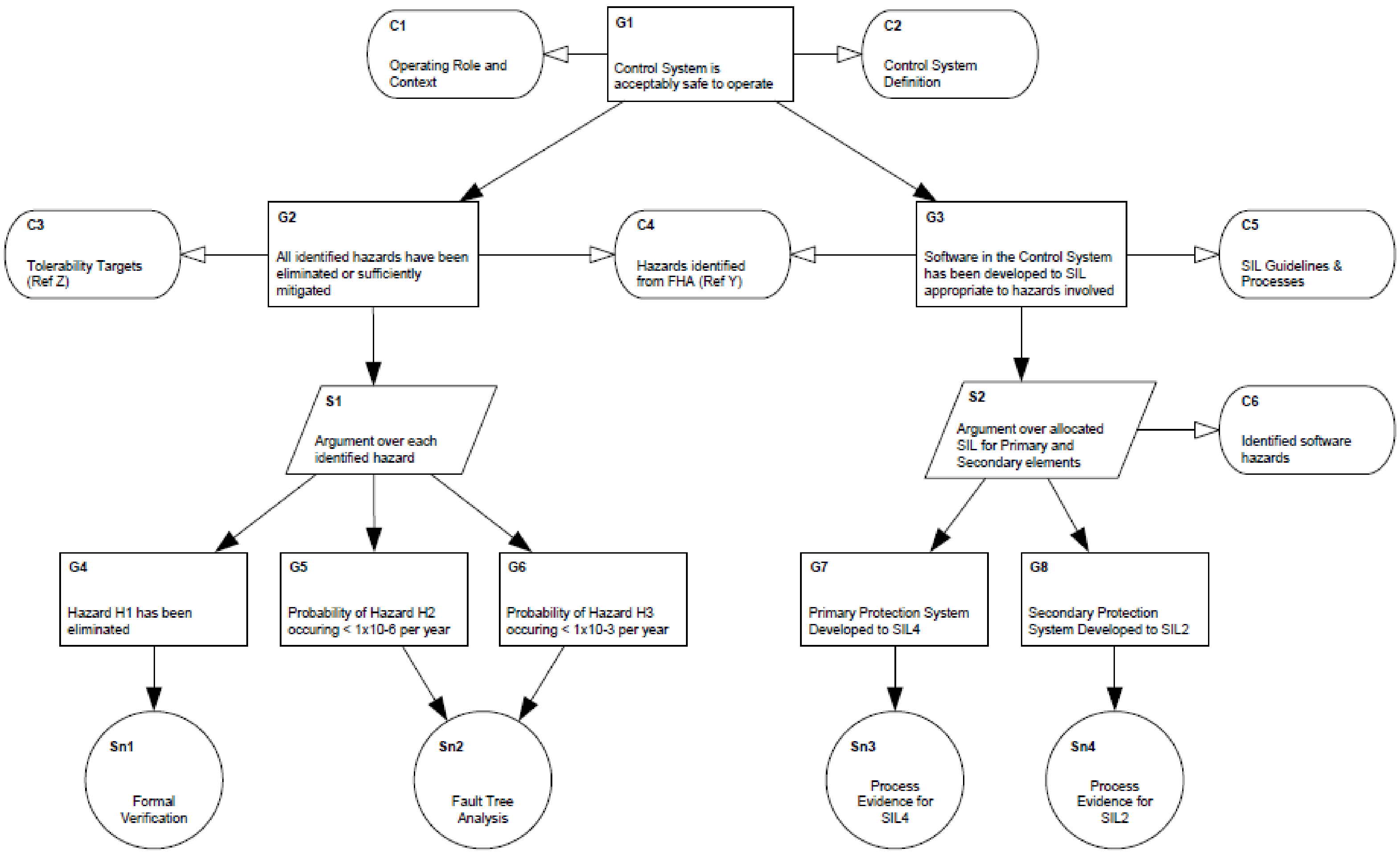
Context

Context in which a claim should be interpreted

Solution

Evidence that supports the truth of a goal

Example Goal Structure


C1
Operating Role and Context
G1
Control System is acceptably safe to operate
C2
Control System Definition
C3
Tolerability Targets (Ref Z)
G2
All identified hazards have been eliminated or sufficiently mitigated
C4
Hazards identified from FHA (Ref Y)
G3
Software in the Control System has been developed to SIL appropriate to hazards involved
C5
SIL Guidelines & Processes
S1
Argument over each identified hazard
S2
Argument over allocated SIL for Primary and Secondary elements
C6
Identified software hazards
G4
Hazard H1 has been eliminated
G5
Probability of Hazard H2 occuring < 1x10-6 per year
G6
Probability of Hazard H3 occuring < 1x10-3 per year
G7
Primary Protection System Developed to SIL4
G8
Secondary Protection System Developed to SIL2
Sn1
Formal Verification
Sn2
Fault Tree Analysis
Sn3
Process Evidence for SIL4
Sn4
Process Evidence for SIL2

Tool Support for GSN

- For small diagrams can get by with ordinary drawing software eg Visio, or even the tools in MS Word), but for full-sized cases need special-purpose tool.
- Commercially supported 'ASCE' tool from Adelard (http://www.adelard.com/web/hnav/ASCE)

Security Assurance Cases

- Can assurance cases be applied beneficially to the security domain?
- No widespread application of security cases in practice, just small-scale case studies and promising experiments
- Some published guidance (Goodenough, Lipson; USA “Building Security In” Initiative)
- SAFSEC standard (Altran Praxis UK) – an approach that unifies safety and security cases.
- IAWG (Industry Avionics Working Group) – also looking at safety/security unification

Key questions

- There are a number of differences between safety and security which have implications for how we might create security cases
- Are these insurmountable?
- Do they undermine the value of assurance cases in the security domain?
- What are the potential benefits?

Safety/Security similarities

- Both can use cause-effect models (fault trees / attack trees)
- Both can derive requirements to mitigate problems identified
- Both can argue in assurance cases that those requirements are implemented and achieve the required mitigation

Safety/Security differences

- Safety
  - Random behaviour of non-human, non-goal seeking world (if a control valve fails it does not actively encourage another to fail)
  - Adaptive behaviour by humans (eg seeking to make their job easier; but may ‘undo’ a bad thing)
- Security
  - Dealing with agents whose goal is to compromise a system
  - May systematically probe a system for vulnerabilities and having breached one defence move on to the next, whereas non-human phenomena do not normally respond to feedback.
  - ‘Hardening’ – security measures that are not responses to specific threats; less common in safety world as additional functionality may increase likelihood of failure.

Other practical issues

- Often hear that safety problems are usually the result of poor requirements/design, whereas security vulnerabilities most commonly due to implementation errors.
- May simply be that the security domain has a lower level of maturity in development processes
- Poor process can make it difficult to produce an assurance case at all.

Report’s Conclusion

- Report concluded that there is “*likely to be incidental rather than fundamental differences*” in the application of assurance cases to the security domain
- Recommended that we experiment with their use
- Outlined some potential benefits

Potential benefits (1)

- Handle complexity – aid to understanding complex arguments and help to assign evaluation work packages amongst a team in an efficient way. Focus attention on critical parts of the system.
- If vendors supply the assurance cases then they provide a summary of the vendor’s security analysis.
- Identify the complete range of evidence available and areas where we may lack expertise to make an informed judgement.
- Help to justify evaluator’s position when requesting a vendor perform re-work or provide additional information.
- Modular assurance cases can promote re-use (eg security case for a particular operating system).

Potential benefits (2)

- Useful ‘security patterns’ in argumentation may emerge
- Corporate memory
  - Product upgrades require us to be able to re-visit the assessment of an earlier evaluator
  - Evaluations often set precedents or policy a bit like case-law.
- Training
  - How does a new evaluator gain insight into the thought processes of an experienced evaluator?
  - What are the thought processes of an experienced (or good) evaluator?

Case Study & Workshop

- Report included a small case study: a secure external keyboard and display for a “Redberry” mobile device.
  - Generated attack trees, security requirements and then presented a GSN-based assurance case
- One-day workshop with representatives from our crypt evaluation community
  - Generally well received
  - Plan for some evaluators to attend a training course in the Adelard ASCE tool
  - Wait and see if they decide to take it further

Outputs

- "Security Assurance Cases: Motivation and the State of the Art", University of York High Integrity Systems Engineering Group, report number CESG/TR/2011/1, 18 March 2011, by Rob Alexander, Richard Hawkins, Tim Kelly
- Case study: Secure Redberry Keyboard and Display Unit.
- Not on York Univ. website, will consult with CESG release manager to see if it can be made public.