

# Simpler Security Targets

Mike Nash

Gamma Secure Systems Limited

<http://www.gammasl.co.uk/>

# What is the problem?

- Consider a developer's view
- Low Assurance Evaluations
  - High Cost
  - Excessive preparation time
  - Evaluator work seems misdirected

# Typical Complaints

- The ST evaluation took longer and cost more than the TOE evaluation
- The ST evaluation didn't find a single flaw in my product or my analyses, only in the documentation you made me write just for use by the evaluators

*Not all ST evaluations  
end up like this*

# How did we get into this state?

- Historically, evaluation methodologies were driven by the goal of high assurance
  - (for low assurance you leave out...)
- Why is there nothing you do for low assurance that isn't done for high assurance?

# Criteria are designed for whose convenience?

- Historically, evaluation was a free service
- Criteria were designed to minimise the evaluator's work

# This paradigm is out of date

- Sponsor now pays all costs
- Typically, developer time costs more than evaluator time
  - In particular work done by developers purely for evaluators has zero product improvement benefit
- But are the CC still designed for evaluator convenience?

# CC Trial Version 2.4

- Proposal – Low Assurance Security Targets
  - Limited to EAL1
    - actually could be used at higher levels
  - No security problem definition or security objectives
  - No need to justify unsatisfied dependencies

# V2.4 Low Assurance Targets

- Reduce ST validation work
- But eliminate much of the usefulness of the ST in security solution validation
- Doesn't matter in all cases:
  - e.g. “designed to meet PP04-04”
  - e.g. procurement specification
  - Some consistency issues remain

# Gut Feeling

- *Still an evaluator driven approach*
- *And how many developers would accept EAL1 as a credible assurance target?*

# Another Current Example

- TOE Summary Specification vs. Functional Specification
- CC Version 2.1 treats as unrelated
- CC Trial Version 2.4 recognises that the TOE Summary Specification is a functional specification of the security functions!

# Possible Solutions

- Solution 1 – eliminate duplication
- Solution 2 – add extra consistency checks
- Guess which solution Version 2.4 chooses
  - Even at EAL1!

# Security Functionality

- ADV components need to confirm claimed security functionality is actually present
- This is done by consistency checking
- Checking could be against
  - A higher level design representation
  - The functional specification
  - The TOE summary specification
  - SFRs

# CC Approach

- CC evaluator checks against SFRs
- Was this the best solution?

# Problems

- Low level design is never checked for consistency against the high level design
- High level design is not checked against the functional specification
- Functional specification is not checked against the TOE summary specification
  - CC Trial Version 2.4 adds this last check

# An alternative approach

- Assertion: SFRs are a semiformal representation of security functionality
- Question: Why require SFRs before you have other semiformal representations?

# Outline alternative ST criteria

- Keep Security Problem Definition and Security Objectives
- Recognise that in some cases security objectives are not derived from a clear security problem
- Link TOE Summary Specification back to Security Objectives

# Use of SFRs

- SFRs become semiformal representation of TOE Summary Specification
- And are checked as such
- Perhaps SFRs become part of the semiformal security functional specification requirements (ADV\_FSP.3)

# Alternative TOE evaluation

- ST evaluation confirms TOE Summary Specification meets security objectives
  - EAL1 – confirm that functional specification matches TOE Summary Specification
  - EAL2 – confirm that high level design matches functional specification
  - EAL3 – look for “completeness” (no side effects)
  - EAL4 – check low level representations

# What about SFRs?

- Somewhere around EAL5 bring in SFRs and semiformal design specifications

# Benefits

- At lower assurance levels, much less CC formalisations and evaluator checking against them
- Understandability, speed, convenience

# Drawbacks

- The equivalent of the current EAL4 (in particular) is devalued
- Although the design and functional specification at EAL4 are informal, perhaps SFRs capture more aspects of the security requirements than a TOE Summary Specification, and this can be checked by evaluation...

# Summary -1

- We want to simplify STs and reduce ST evaluation effort
  - Actually we want to simplify STs used in low assurance evaluations
- CC Trial Version 2.4 has an approach for EAL1, but it's really only suitable for limited types of evaluation

# Summary - 2

- CC Trial Version 2.4 adds other things that push up ST evaluation effort
- If SFRs are a semiformal representation, then perhaps they ought to appear around EAL5
- Actually, we only need a TOE Summary Specification to perform a CC evaluation

# Summary – 3

- If our evaluation paradigm is wrong, we will specify poor (inefficient, redundant) evaluation criteria
- Starting with ST evaluation
- Is this why evaluation is not more used?

# Simpler Security Targets

Mike Nash

Gamma Secure Systems Limited

<http://www.gammasl.co.uk/>