



Evidence Based Approach

Shaun Gilmore, US Scheme

ICCC, Jeju

9/23/2008



Outline

- Current Issues
- Proposal
- Working Group Actions



Outline

- Current Issues
- Proposal
- Working Group Actions

Current Issues

- Current paradigm embodied in the CC is that of a “top down” or “waterfall” development model
 - one activity leads into the next
 - each subsequent activity producing a more detailed abstraction
 - Traditionally HLD → LLD → IMP
- Market pressures no longer allow a waterfall type development process
 - Too time consuming
 - No payoff
 - Rapid/Iterative/Spiral Development Models

Current Issues

- The perception is that CC requires evidence in a particular form
- We continually see evidence being produced for the sole purpose of evaluation
 - Often third party produced
 - Time consuming, Costly, Inaccurate
 - Actually detracts from product assurance
 - Resources can be utilized on development assurance activities

Current Issues

- Lower assurance levels focus too much on design rather than implementation flaws
 - CC is routinely criticized for not focusing on the vast majority of real world vulnerabilities.
- This approach may also be true for large complex products

Current Issues

- We simply cannot look at everything!
 - Number of products entering evaluation is growing faster than resources available
 - Product complexity is growing beyond the capabilities of scalable human analysis

Current Issues

Large Products – Can we achieve medium or high assurance?

- How do we enumerate every interface?
- How do we test the entire product?
- How do we understand the interaction and dependencies on the environment and underlying platform?
- Can we define a complete and accurate architecture diagram and description?
- What does it mean to be complete?
- When is a sample size of sufficient size to be adequate?



Outline

- Current Issues
- **Proposal**
- Working Group Actions

Proposal

- More closely align current evidence requirements with actual development process documentation
- Relax and/or eliminate existing assurance requirements to those that add value
 - Focus less on formality of the form of the information but rather on the value of the content
- Provide a minimal baseline of necessary evaluation evidence
 - *Not adequate to assess solely what is produced
- Provide guidance as to acceptable forms of evidence
- Provide a framework for better evaluator/developer interaction

Proposal

ADV_ARC

- **Developer actions:** The developer must design and implement the TOE such that it: cannot be bypassed; protects itself from untrusted entities; and isolates resources to be protected such that all interactions are controlled.
- The developer must provide an architectural diagram(s) depicting the TOE security architecture – to include the trust boundaries – and a written description of the TOE security architecture.
- **Content:** The architectural diagram and description should combine to provide an understanding of what the TOE security architecture is and how it works to meet the selfprotection, nonbypassability and resource isolation requirements.
- The level of detail should be consistent with the level of architectural rigor claimed for ADV_INT.
- **Evaluator actions:** Verify the suitability of the security architecture and that the content of the evidence is sufficient to provide a high level understanding.
- Evaluators will interact with developer staff to fill any holes in understanding the security architecture design and implementation

Proposal

- Explore replacing or supplementing the existing assurance evidence paradigm of the CC for lower assurance levels
- Create a methodology that focuses more on common vulnerabilities rather than design documentation
 - Even if we eliminate the most common vulnerabilities it will be a measurable improvement

Proposal

- Develop TOE Development Process assurance requirements to help mitigate implementation flaws and to better enable “Predictive Assurance”
- Perhaps all we can do at this stage for large products
 - Gain assurance and confidence in development processes

Proposal

ADV_TDP – TOE Development Process

- Developer must provide a description of the development process (to cover design, implementation, testing, maintenance, etc.) of the TOE. Requirement families for:
- Process Assurance activities (e.g., threat modeling, design and implementation effectiveness, development process controls, change analysis, development environment security, related item consistency controls)
- Use of Automation (e.g., design and implementation analysis tools, release validation tools, configuration management tools)
- Testing Activities (functional and penetration – done by developer, security standards compliance)
- Flaw Remediation (reporting support, flaw analysis processes, remedy distribution procedures, vulnerability reporting)



Outline

- Current Issues
- Proposal
- Working Group Actions

Working Group Actions

- **Only in its infancy and we understand this is a large undertaking and a lot of work needs to be done**
- Develop a methodology to assess developer tools and processes for lower assurance evaluations and to serve as the foundation for complex products
- Work closely with the Predictive Assurance working group to establish evidence requirements for development processes that meet overall objectives
- Develop a consistent bar on minimal acceptance for evidence among all schemes
- Review CC requirements to relax documentation requirements and provide guidance where appropriate
- Work with vendors, labs, and customers to ensure a practical and feasible result



Evidence Based Approach

Questions