Common Criteria Version 4

Proposals for New Evaluation Approaches

Anthony Apted and James Arnold

September 23, 2008

Energy | Environment | National Security | Health | Critical Infrastructure

SAIC
From Science to Solutions

Synopsis

- Considerations for Common Criteria (CC) Version 4 (V4)
- Rationale for proposed approaches
- Security target
- Design evidence
- Guidance documentation
- Life-cycle evidence
- Testing
- Vulnerability analysis
- Evaluation outputs
- Alternative assurance levels
- Conclusions

Considerations for CC V4

- Address CC Version 3 (V3) goals:
  - Eliminate redundant activities
  - Reduce or eliminate activities that contribute little to assurance
  - Clarify CC terminology
  - Restructure activities to focus on areas where assurance is gained
  - Add new requirements as needed
- Avoid CC V3 errors
- Protect developer investment in CC
- Acknowledge role of consultants

Rationale for Proposed Approaches

- Counter accusations that CC evaluation:
  - Evaluates the documentation, not the product
  - Is a mechanical exercise in checking off requirements
  - Does not add assurance in the security of the product
  - Does not produce meaningful results
- Consider approaches that reduce burden on developer to produce evidence specifically for the purpose of CC evaluation
- Reconsider assurance requirements that add little or no actual assurance
- Evaluation evidence categories:
  - Purpose of evidence?
  - Contribution to assurance?
  - Product of development or CC-specific?

Security Target

- Produced solely for CC purposes
- Described as top-down specification
- Developed as bottom-up description
- Changes during course of evaluation

Proposal: Evaluators write security target (ST) in conjunction with developer

- Initial draft forms the agreed basis for evaluation
- Final version is accurate statement of what was evaluated
- ST becomes an evaluation output
- Final ST evaluated by validators or certifiers

Design Evidence

- Enables evaluators to understand Target of Evaluation (TOE) and its security functions
- Facilitates evaluator functional and penetration testing
- Provides assurance in correct implementation of Security Functional Requirements (SFRs)
- Contributes to understanding TOE self-protection
- CC V3 requirements unlikely to be satisfied by standard developer evidence

Proposal: Do not evaluate against Pass/Fail criteria

- Evaluators use whatever developer has available or is willing to provide
- Evaluators develop own design representation
- Evaluators can work with available consultants
- Requirements specify what evaluators need to understand about TOE

Guidance Documentation

- Describes how TOE users handle TOE securely
- Guidance documentation is part of TOE
- Does not contribute to assurance

Proposal: Only requirement should be that guidance describes how to install, manage and use TOE consistent with ST

  - Inaccuracies have to be corrected in documentation (no addenda, *readme* files, etc.)
  - Standard means to identify relevant evaluated guidance documentation

Life-Cycle Evidence

- Describes procedures supporting TOE development
- Mature procedures contribute to product quality
- CC-conformant descriptions of procedures do not contribute to product quality
- Developers have procedures, but not documented to CC standard

Proposal: Evaluation team assesses procedures and processes, whether documented or not

  - Developer can provide documentation, but is not compelled to do so
  - If documentation is available, evaluators assess procedures against documentation
  - Otherwise, evaluation team conducts study, obtaining information from whatever sources are available
  - Evaluation team documents approach and findings, with assessment of maturity and durability of evaluation results

Testing

- Test evidence provides indication of developer testing effort
- At lower evaluation assurance levels (EALs), developer not required to perform comprehensive testing
- Most developers perform some testing, although geared to product capability, not security functionality
- Developers often create new test suites specifically for CC

Proposal: Evaluation team develops and conducts tests appropriate for the evaluation

  - Developer chooses to provide test documentation or describe approach to testing and bug handling
  - Evaluation team assesses developer's test regime and produces coverage and depth analyses
  - Evaluation team is free to use any developer test support, but ultimately must identify or develop an adequate set of security tests

Vulnerability Assessment

- CC V3 removes requirement for developer to produce vulnerability analysis documentation

Proposal: Remove link between requirement level and attack potential

- Evaluation team devises and conducts penetration tests based on understanding gained of TOE
- Evaluation team reports evidence available and effort expended in functional and penetration testing
- Consumers and schemes derive idea of attack potential to which TOE was subjected during evaluation, or otherwise develop idea of level of assurance obtained

Evaluation Outputs

- CC criticized for not producing useful results
- But what constitutes useful results?
- This seems to be an issue for individual schemes
- Schemes should identify needs of customers and define useful evaluation results

Proposal: Evaluation team responsible for a broader set of published evaluation results

  - As previously identified, evaluation team writes ST as evaluation output
  - Evaluation team explicitly identifies the guidance appropriate for use of the evaluated product
  - Evaluation team produces a nonproprietary test report describing evaluation team test effort and tests performed

Alternative Assurance Levels

- Proposed approaches may not fit with the current CC model of hierarchical assurance

Proposal: Alternative assurance levels can be used to further qualify evaluation assurances

- ***Developer Assisted***: Developer provides whatever documentation is available, but does not produce new documentation specifically for evaluation (equivalent to EAL1–EAL4)
- ***Developer Demonstrated***: Developer provides semi-formal design documentation, process documentation, and test documentation (roughly EAL5)
- ***Developer Verified***: Developer provides formal design documentation, process documentation, and test documentation (roughly EAL7)
- These classifications acknowledge that a developer can provide evaluation-specific evidence and gain credit for doing so

Conclusions

- We have made a number of proposals for approaches to evaluation
- The proposals were developed with the following goals:
  - Increasing assurance in the product that an evaluation should deliver
  - Easing the burden currently placed on developers that undertake CC evaluations

Contacts

Anthony Apted

SAIC Accredited Testing & Evaluation Laboratories

Common Criteria Evaluator

anthony.j.apted@saic.com

James Arnold

SAIC Accredited Testing & Evaluation Laboratories

Technical Director

james.l.arnold.Jr@saic.com

http://www.saic.com/infosec/common-criteria/