High Robustness Cross Domain Solutions Tiger Team

John Mildner
Jennifer Guild

Layered Assurance Workshop - 2011

Purpose

Provide an overview of the High Robustness Cross Domain Solutions Tiger Team (HR CDS TT)

Provide status on support to NIST SP 800-53 development

Definitions

- Robustness is the measure of confidence that a system operates as required, designed, and expected throughout its lifecycle ensuring essential services, coping with faults, failures, unexpected interactions and malicious activities
- High Robustness provides the technical infrastructure to enable survivability and mission integrity in high threat environments.
  - For CDS, High Robustness reduces risk associated with information sharing across a wide range of domains
- High Robustness is achieved through design, engineering, and implementation practices throughout the system lifecycle
  - Provides the means to improve current best commercial practice

High Robustness is Critical to CDS –Today's Reality

- Information sharing across domains will only increase
- Threat agents operate within the domains
- Potential for security policy violations with catastrophic results
- System complexity increasing (weakest link paradigm applies)
- Commercial products are typically low or medium robustness
- Desire for network visibility of CDSs (target)
  - Centralization (known locations)
  - Remote management and monitoring
  - Feedback to low-side senders
  - Increasing data complexity and throughput

High robustness will reduce the risk associated with the modern net-centric environment

HR CDS TT Formation

- Tiger Team formed under the Community Security Test Group (CSTG) to understand how High Robustness can reduce security and programmatic risk in emerging technologies
    - Understand high robustness relationship to CDS design
        - Hardware and software implementation
        - Operating Systems (NIAP policy concern)
        - Emerging technologies (e.g., Separation Kernels)
    - Supports validation of advertised capabilities of CDS products
    - Provide community education (developers, evaluators, consumers, integrators, approvers)

Definitions

- System = Foundation + Non-Kernel Security Related Functionality
- The Foundation is the hardware, firmware, and the kernel components that implement a set of security mechanisms only accessible via kernel interface.
- The Non-Kernel Security Related Functionality (NKSR) makes or enforces policy decisions or operate correctly to maintain data correctness and supports either an interface to the foundation or an interface to applications.

Non-Kernel Security Related Definitions

- NKSR can be further refined into functions that interface to the foundation (NKSR-Kernel) and those functions that interface to the applications (NKSR-Application).

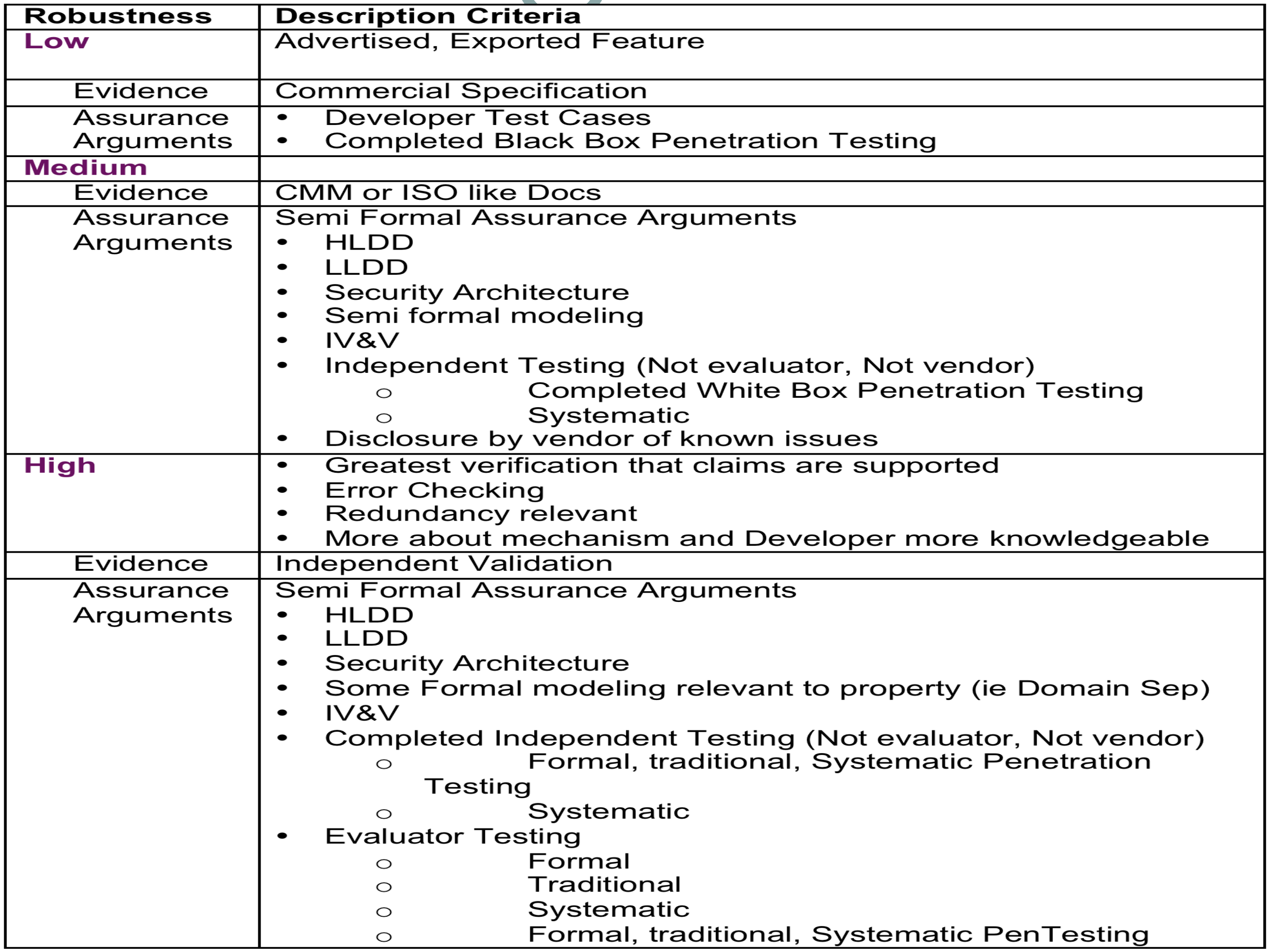
- NKSR-Kernel supports an interface to the foundation and supports/ enforces system security policy or operates correctly to maintain data correctness at the root/admin privilege level.

- NKSR-Application supports an interface to applications and supports/ enforces application policy decisions or operate correctly to maintain data correctness. NKSR-Application may include security-related applications.

Definition of Levels

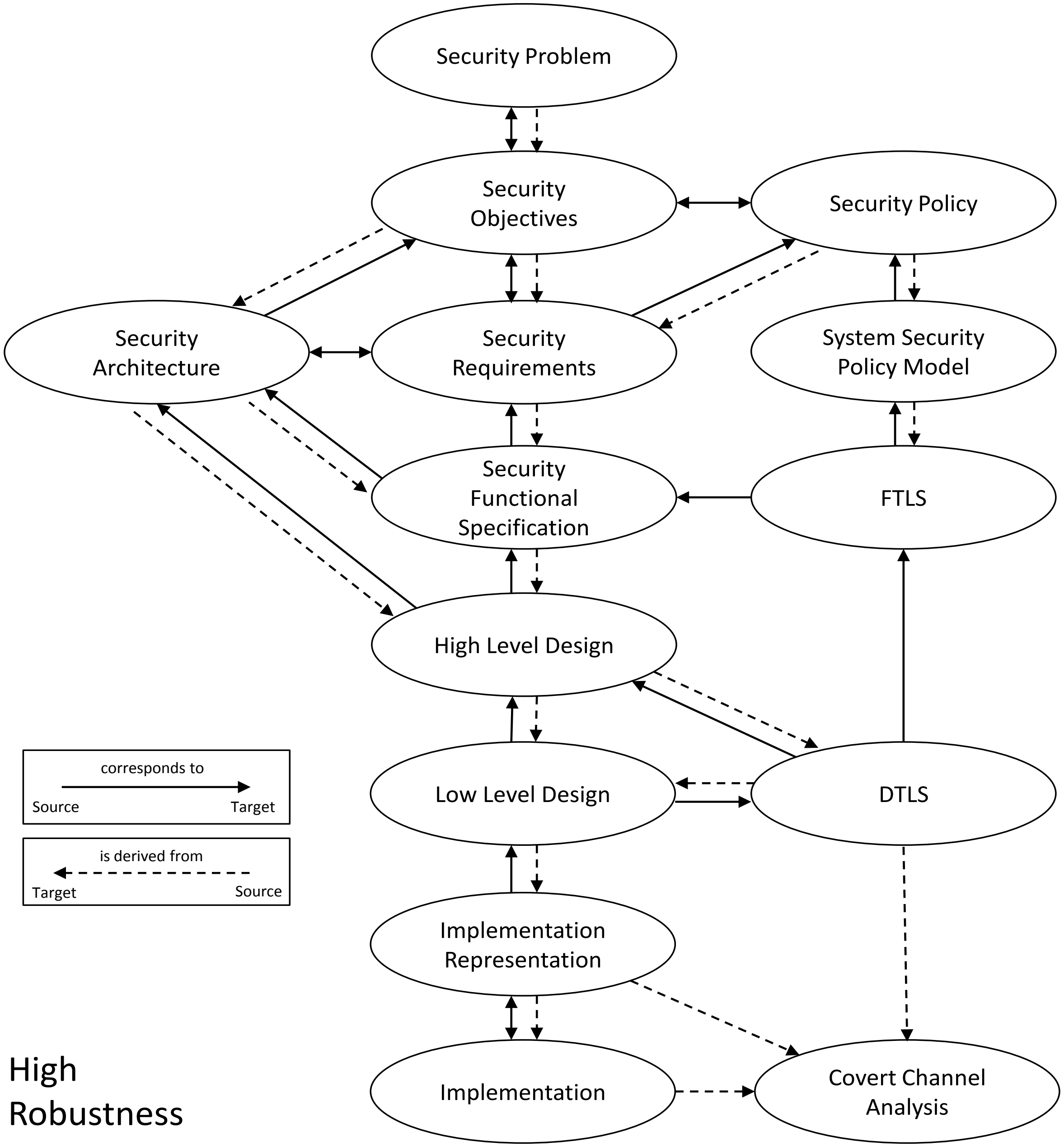
Robustness	Foundation	NKSR-Kernel	NKSR-App
Low	Low Low Low	Low Medium Low	Low Low Medium
Low-Medium	Low	Medium	Medium
Medium-Low	Medium Medium Medium	Low Low Medium	Low Medium Low
Medium-Medium	Medium Medium Medium	Medium High Medium	Medium Medium High
Medium-High	Medium	High	High
High-Medium	High High High	Medium High Medium	Medium Medium High
High-High	High	High	High

Robustness Criteria



Robustness	Description Criteria
Low	Advertised, Exported Feature
Evidence	Commercial Specification
Assurance Arguments	• Developer Test Cases • Completed Black Box Penetration Testing
Medium	
Evidence	CMM or ISO like Docs
Assurance Arguments	Semi Formal Assurance Arguments • HLDD • LLDD • Security Architecture • Semi formal modeling • IV&V • Independent Testing (Not evaluator, Not vendor) ○ Completed White Box Penetration Testing ○ Systematic • Disclosure by vendor of known issues
High	• Greatest verification that claims are supported • Error Checking • Redundancy relevant • More about mechanism and Developer more knowledgeable
Evidence	Independent Validation
Assurance Arguments	Semi Formal Assurance Arguments • HLDD • LLDD • Security Architecture • Some Formal modeling relevant to property (ie Domain Sep) • IV&V • Completed Independent Testing (Not evaluator, Not vendor) ○ Formal, traditional, Systematic Penetration Testing ○ Systematic • Evaluator Testing ○ Formal ○ Traditional ○ Systematic ○ Formal, traditional, Systematic PenTesting

Development Representation



High Robustness

Accomplishments

- Developed draft framework document that provides:
    - Definition of levels of robustness (Medium, Medium-High, and High)
    - Definition of the High Robustness architecture (Foundation, Kernel and Application)
- Identified and defined development assurance artifacts (e.g.):
    - Security Problem and Security Objectives
    - High Level Design
    - Low Level Design
- Identified 800-53 controls essential for operating system evaluation

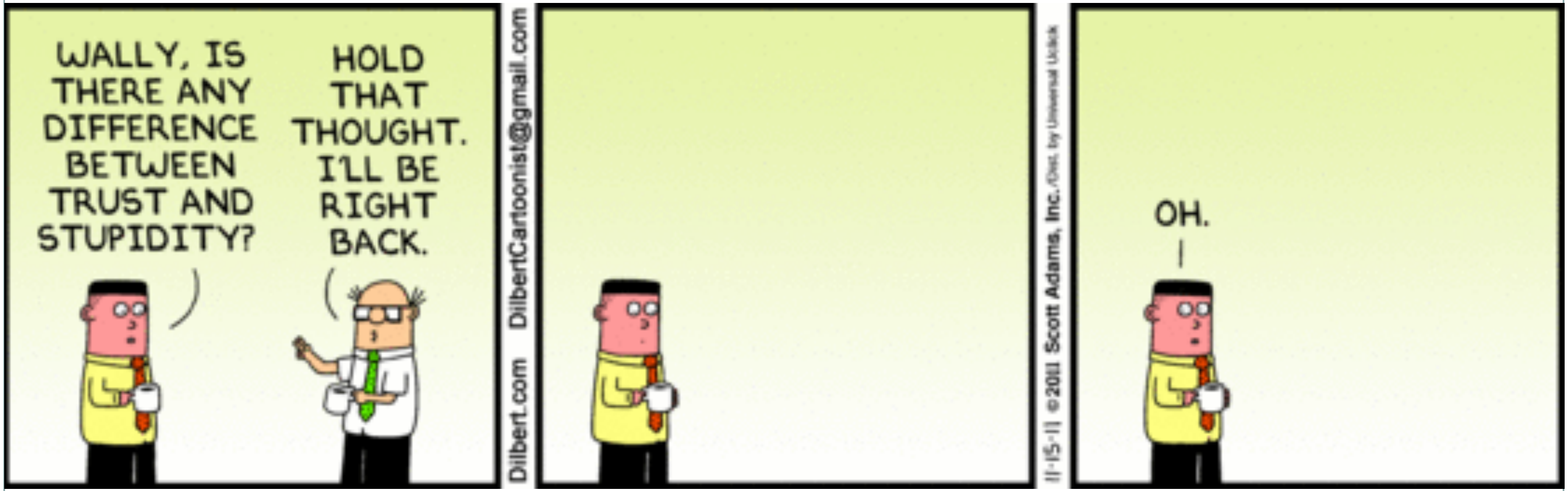
Further Research

- Finalize assurance artifacts:
- Further define high robustness relationship to:
  - Configuration management, static and dynamic code review, supply chain, etc.
- Develop and promulgate community education plan
- Define strategy for efficient evaluation of medium and high robustness CDSs

NIST SP 800-53 Rev 4

- Being led by Ron Ross
- Appendix E being transformed to address “Trustworthiness”
- New Security Controls (such as acquisition artifacts)
- Linkage to SP 800-37, SP 800-39, and future SP 800-xx (Security Engineering)
- Coordination draft expected early 2012.

Questions?


WALLY, IS THERE ANY DIFFERENCE BETWEEN TRUST AND STUPIDITY?
HOLD THAT THOUGHT. I'LL BE RIGHT BACK.
OH.

Dilbert.com DilbertCartoonist@gmail.com

11-15-11 ©2011 Scott Adams, Inc./Dist. by Universal Uclick