Abstract

We have described two complementary activities:

− a MILS Integration Protection Profile, and
− A Common Criteria Authoring Environment (CCAE) to support authors of MILS PPs and STs

Together these provide strategic coordination to the MILS community.

The CCAE will enable authors to produce reviewed PPs and STs of higher quality in less time, and ones that will better serve the common interests of the MILS community.

The CCAE will be a “community source” effort.
Bottom Up and Top Down

- **Existing PPs** are for *components*
- Makes sense to create a marketplace for COTS MILS components
  This is the “bottom up”
- We need to know that when we put them together, we have a *secure* and *evaluatable* system
- The *MILS Integration PP* describes *how* the components come together in a *MILS system*
  This is the “top down”
- We need a practical way for the two to effectively meet
- The *CC Authoring Environment* for MILS connects them
  This is the “meeting in the middle”

This Presentation

- What Common Criteria protection profiles (PPs) do
- Role of the CC PP in product development
- Need for component PP coordination in MILS
- How MILS PPs are developed now
- What’s wrong with it
- How the PP development process can be improved
- A Common Criteria Authoring Environment (CCAE)
- A collaborative environment for MILS specification
What CC protection profiles do: The CC provides us with

- A structure for the development of security requirements specifications
- Independent functional and assurance dimensions (like ITSEC, unlike TCSEC)
- CC introduces Protection Profile concept
  - Remedies problems encountered with ITSEC evaluations
    - Vendor could make claims for any point in the space of functionality × assurance and have those claims evaluated
    - Users were left comparing apples and oranges
  - PPs constrain the space of compliant products
  - PPs are written and evaluated by experts to present a “balanced” set of requirements to developers

What CC protection profiles do: Unconstrained Function × Assurance space
What CC protection profiles do: Function × Assurance space constrained by protection profiles

CC-based product (TOE) development (1/2)

The (logical) sequence of TOE development:

CC → PP_{Type} → ST_{Type} → TOE_{Type}

That’s the development path for One TOE of One product Type
CC-based product (TOE) development

(2/2)

We expect multiple TOEs of each product type

MILS is based on multiple cooperating products defined by related Protection Profiles

- MILS Integration Protection Profile (MIPP)
- Separation Kernel (SKPP)
- Partitioning Communication System (PCSPP)
- MILS Console System (MCSPP)
- MILS Network System (MNSPP)
- MILS File System
- . . .
Can Our PPs Achieve This Goal?

Need for MILS Integration PP and component PP coordination
Many real systems will have more MILS Components

- MILS promises to provide high-assurance security in a broad range of contexts from embedded systems to workstations and servers

- Some prospective customers have multilevel security (MLS) requirements

- To implement general purpose MLS on MILS will require even more components than the few pictured here

What we need for a high-assurance MILS-based MLS Workstation . . .* 

- We’ve got MILS SK, PCS, DDS, CORBA, guest OS, POSIX . . .
- We’ll need other high-assurance MILS subsystems:
  - Console with trusted window system
  - Trusted disk storage
  - Trusted networking
  - Session management
  - Application management
  - MLS objects, attributes and policy arbiter (RVM)
  - User I.A & A
  - Cryptographic support
  - System self-test, integrity and recovery
  - System management
  - Auditing

*The real list may have upwards of two dozen PPs to write, review, integrate and evaluate.
If One Vendor Could Do It ALL

We should expect:

System A
- SK
- PCS
- Console
- Network
- File System

System B
- SK
- PCS
- Console
- Network
- File System

! = It works!

But are any of us intending to do it all? I don’t think so! Not anytime soon.

MILS architecture is based on composition

- A “double-whammy” of high assurance and composition
- Components developed by different vendors
- Components are defined by Common Criteria-style protection profiles (PPs)
- The collection of PPs reflects an intended architecture
- The PPs must be in agreement with the architecture
- So how do these PPs come about?
How MILS PPs Are Written Now*

Existing PP Examples (not always good*)

*Produce a PP for Something*

CC v??

[Diagram: CC v??, ST process, PP for Something, Review Cycle(s), Domain Expertise + Security Expertise (but not always...*)]

* No offense intended

MILS PP author’s dilemma

- Make literally thousands of interdependent decisions
- Produce a document of legal precision
  - Level of complexity and interdependence makes it like programming
  - Unlike programming, there’s no way to test it
- Possibly no specific training, method or tools
- With no examples (or poor ones)
- And little (or no) other guidance
  - PPRB only involved in “US Government” PPs and do not seem to apply any modern automation or collaboration technology
- Must enable your competitors if writing a PP for your own product

- It’s like
  - A pilot without a checklist
  - A captain without a compass
  - A surgeon without a procedure
  - An accountant without a spreadsheet
  - A construction worker without power tools
What’s wrong with this?

- It takes a long time (2+ years) and a lot of effort ($$$)
- Very tedious and error prone work
- Bad examples are propagated like a virus
- Difficult to understand and to track CC versions
- Difficult to assess impact of global change to MILS PP family
- Difficult to generate and maintain mappings
- Difficult to check consistency and completeness
- Difficult for document to feed into further development
- Authors may not have expertise in CC or security
- PP and ST authors have little guidance or ability to enforce / achieve shared standards
- PP and ST authors are underconstrained to achieve goals
- Requires “legal” precision of language unfamiliar to some
- Little support to structure the author’s efforts
- Nothing to assure that the PPs will “hang together”

What would help?

- A god-like security architect with infinite attention to detail and the ability to keep an infinite amount of stuff in her head,
- …could write the PPs and STs for all the MILS components, ensure their consistency and completeness, and hand them to the developers on stone tablets.
- We don’t have that.

So how about some help for mere mortals?
- Help with the detail bits
- Help with all the stuff you have to know and remember
- Help with the myriad decisions
- Help with the consistency and completeness
- Help with how to go about it
- Help to assure composability, compositionality and interoperability
Logical structure of a protection profile superimposed with a security taxonomy

Taxonomy yields better conceptual coverage:
External Factors x High-level Rqmts x Phases => Properties x Deps => Rqmts

A CC Authoring Environment for MILS -- What it would provide (1/2)

- Common Criteria in a structured, machine (re)usable form
  - CC model and conventions as described in CC Part 1
  - Component operation application rules
  - CC SFR catalog of Part 2 and SAR catalog of Part 3
  - A “Plugged-in CC”, instead of “CC Unplugged”

- Library of documentation conventions and style sheets
  - Foundation style sheet object classes
  - PP style sheet object
  - ST style sheet object
  - Formatting and typography rules

- Catalog of (re)usable community standards:
  - Definitions of basic CC and MILS terms
  - MILS evaluator guidance and robustness level guidance
  - Bibliography of MILS references
A CC Authoring Environment for MILS -- What it would provide (2/2)

- Mechanical checks
  - Consistency
  - Constraints needed for composability and compositionality
  - Requirements traceability
  - Statistics
- Guidance based on expert knowledge base that can evolve and be adapted.
  - Security ontological structure (expanded taxonomy with semantics)
  - Security analysis workflow
  - Expert usage / instantiation patterns
  - Decision support
  - MILS architecture components, relationships and constraints
  - PP and ST development work flow rules
  - CC presentation conventions
  - Guidance for desired robustness level
  - Evaluator guidance
- Output that can be (re)consumed by these or other tools

A CC Authoring Environment for MILS -- Benefits (1/2)

- Achieve uniformity and sufficiency of PPs and STs
- Relieve much of the tedium, to better apply author’s effort
- Reduce/eliminate many types of errors and inconsistencies
- Reduce the document maintenance problem
- Shorten PP and ST development time and raise quality
- Can be used by authors and reviewers of PPs and STs to explore/query the information represented in the document
- Explore / create “what if” variants
- More easily adapt to later versions of the Common Criteria
- More easily incorporate evolving community standards
- More easily revisit existing PPs and STs when security environment or external requirements change
A CC Authoring Environment for MILS -- Benefits (2/2)

- MILS PPs harmonized to achieve additivity property for foundational PPs
  - Can be realistically used together
  - No gratuitous entropy among the PPs
- Expert knowledge base can grow, adapt, come from new sources, and be refined and effectively be passed on to others
- Automated repeatable checking encourages continuous QA
- Produce a database representing the current stage of product definition that can be input to the next stage (e.g., PP --> ST --> ... )
- Produce output that can be consumed by other tools during product development
- Provide a vehicle for applying / propagating the MILS Integration PP constraints to all MILS component PPs and guaranteeing coherence
- Help ensure that the PP or ST remains a living part of the definition and development of a product

A CC Authoring Environment for MILS -- What it is Not

- Not a pushbutton protection profile
  - "Protection Profiles for Dummies" - not!
  - Substitute for a knowledgeable author - not!
  - It IS a power tool for subject matter experts
- Not a simple “template” for a protection profile
  - (A template would result from removing from one PP everything that makes it different from other PPs)
    - wouldn’t leave much more than a top level outline and a set of paragraph styles in MS Word
    - may save the author a little bit of document setup in MS Word
    - wouldn’t help with the hard stuff
  - It IS more like a class library, with inheritance, that must be instantiated and specialized for a particular PP
CCAE Concepts

- Libraries
  - Environment library
    - Common Criteria corpus
    - CIMs, Interpretations
    - MILS / Security ontology
    - Resource registry
    - MILS Integration Framework
  - Document generation library
    - Document component classes
    - Document generators
    - Instantiation and rewrite rules
  - Rule-base library
    - Expert knowledge
    - Common Criteria process and convention rules
    - Work flow rules
    - Relational model
    - Computational rules for consistency checks, statistics, type checking, query templates

- Authoring framework
  - Negotiation model
  - User interface
  - Query processor
  - Inference engine
  - Rule execution
  - XML generation
  - Advice generation
  - Algorithms: composition, goal-driven negotiation, fuzzy unification, heuristics

- Document converter / renderer
  - I/O, XML import / export
  - Generate .pdf, .docx, .xlsx, etc.

- Document repository
- Current document factbase

Environment Library

- “Plugged-In” Common Criteria, by versions
  - Lifetime of last official version, 13 months (proves the point!)
  - New CC version 3.1 available in XML
    - CC parses into Prolog terms with existing SGML / XML parser
    - Can proceed immediately to create access predicates
    - Build relations within the CC, e.g., dependencies, EALs, custom EALs
    - Path references back to text in XML for display and export
    - Relations to MILS ontology and expert knowledge
    - Huge saving of effort to achieve a “plugged-in” CC
  - Older versions will require some labor

- MILS technology and security ontology
  - OWL (Ontology Web Language) library for Prolog
  - Create a consistent and semantically rich representation of security threats, policies, assumptions, objectives, functional countermeasures, and assurance measures
  - MILS conventions and standards
  - Flow-down constraints from MILS Integration PP
Document Generation Library

- XML DTDs and stylesheets
- Template class library
  - Structured PPs, STs, and other documents (FSP, …)
  - Documents constructed from component classes
- Template instantiation and rewrite rules
  - Describe the valid combinations of components from class library
- Other useful resources
  - EAL packages (CC standard EALs, and med- and high-robustness)
  - CC document construction rules

Rule-Base Library

- Processing rules
  - Common Criteria process and conventions
  - Work flow
  - Query processing
  - Computation of consistency checks, statistics, dependency checks, EAL conformance checks
- Compositional rules
  - Allocation of requirements and policy responsibility
  - Based on sound mathematical models
  - Guided by MILS Integration PP
- Expert knowledge
  - Applicable countermeasures for given threats / objectives
  - Application recommended patterns
  - Evaluator recommendations and evaluation methodology
Expert Knowledge

- PP authors may not be security experts and/or may not have written a PP before
- We would like to effectively bring to the author the knowledge of experts:
  - Security engineering
  - Evaluation requirements and methodology
  - Academia and security research
  - Common Criteria model, methodology, and documentation
  - MILS architecture
- Evolving and improving on an on-going basis
- Distributed and applied by authors as quickly as possible

CCAE Collaboration

CCAE Collaboration Environment

- PP
- ST
- CCAE
- Reviewers
- Certifiers
- Author
- Evaluators

Rance DeLong and John Rushby

34
CCAÉ Prototype Development

- Libraries
  - CC "plugged-in"
  - Ontology / Taxonomy
- Prolog (logic programming language)
  - Rapid prototyping, declarative language
  - Flexible relational database
  - Turing-complete query language
  - Built-in inference engine, unification
  - Describing and processing other languages: DCGs
  - Modern document support: SGML / XML parser / writer library
  - Ontology support: OWL (Ontology Web Language) library
  - Foreign language interfaces
  - Ideal vehicle for prototyping the CCAÉ

Relational Model of a PP (1/5)

Let
\[ T \] the universe of threats
\[ \Pi \] the universe of organizational policies
\[ A \] the universe of assumptions
\[ \Omega \] the universe of security objectives
\[ SFR \] the universe of CC security functional requirements
\[ SAR \] the universe of CC security assurance requirements
Relational Model of a PP (2/5)

- The PP author defines the security environment SE
  \[ T_{SE} \subseteq T \]
  \[ \Pi_{SE} \subseteq \Pi \]
  \[ \Lambda_{SE} \subseteq \Lambda \]
  derives the security objectives of the protection profile
  \[ \Omega_{PP} \subseteq \Omega \]
  the security functional and assurance requirements of the PP
  \[ SFR_{PP} \subseteq SFR \]
  \[ SAR_{PP} \subseteq SAR \]
establishes the relations
  - Threat-to-Objective \( \subseteq \) \( (T_{SE} \times \Omega_{PP}) \)
  - Policy-to-Objective \( \subseteq \) \( (\Pi_{SE} \times \Omega_{PP}) \)
  - Assumption-to-Objective \( \subseteq \) \( (\Lambda_{SE} \times \Omega_{PP}) \)
  - Objective-to-SFR \( \subseteq \) \( (\Omega_{PP} \times SFR_{PP}) \)
  - Objective-to-SAR \( \subseteq \) \( (\Omega_{PP} \times SAR_{PP}) \)
and the rationale for the relational elements
  - Associate with each element R of the relations above, R x E
  - Where E is an explanation

Relational Model of a PP (3/5)

- The \( \Omega \)-anchored space PP of tuples
  \[ PP = (2^T \times 2^{\Pi} \times 2^{\Lambda} \times \Omega \times 2^{SFR} \times 2^{SAR}) \]
  represents all possible PP relations
- The relation E:
  \[ E \subseteq (2^T \times 2^{\Pi} \times 2^{\Lambda} \times \Omega \times 2^{SFR} \times 2^{SAR}) \]
is an oracle accepting “evaluatable” PPs
- The relation M \( \subseteq \) E is an oracle accepting evaluatable MILS PPs
Relational Model of a PP (4/5)

- The PP consists of descriptive and normative sections
  - Descriptive sections include, “TOE Description,” “Rationale” and the appendices
  - For some set $M^\dagger$ of concrete ground terms:
    \[
    M^\dagger \subseteq M
    \]
    $M^\dagger$ captures the essence of the normative sections of an evaluable MILS PP
  - We’d like to know whether a candidate PP $M_C$ is such a $M^\dagger$ but can’t decide that because we don’t know $E$ or $M$ a priori but we can devise an approximation $M_{CCAE}$ of $M$
  - We can ask whether $M_C$ is consistent with $M_{CCAE}$, and to what degree
  - We can also measure coverage of $M_C$ with respect to $M_{CCAE}$

Relational Model of a PP (5/5)

- Automated consistency checking and guidance
  - A set of instances $m^E \in M^E$ describe typical patterns of usage of the elements as determined by experts
  - For a putative MILS PP $M^\dagger$ with components $m^\dagger \in M^\dagger$
  - For an $M^\dagger$ component $m^\dagger = (t, p, a, \omega, sfr, sar)$ where $\omega \in \Omega_{pp}$, we say that $m^\dagger$ may be fuzzily unified with $m^E$
    \[
    m^\dagger \approx_F m^E
    \]
    if a threshold measure of similarity is achieved
  - Discrepancies in $\approx_F$ expose differences with respect to the expert pattern(s)
  - Ontology used to detect redundancies / misuse of concepts
  - \{Ontology $\oplus$ $M^E$ $\oplus$ Rules\} $\otimes$ $M^\dagger$ $\Rightarrow$ Expert Guidance
Expert Guidance and Advice (1/3)

- The concept:
- Knowledge acquisition
  - Explicit rule encoding
  - Generalization from expert interaction on specific authoring projects
  - Harmonization of knowledge from different experts
- Knowledge application
  - Expert patterns constructed from expert knowledge base
  - Author patterns are constructed from the draft PP
  - Author patterns are “compared” to expert patterns
  - Advice is generated for the author’s consideration
- Negotiation model of interaction
  - author and system negotiate an acceptable PP

* fuzzy unification

Rance DeLong and John Rushby

Expert Guidance and Advice (2/3)

A simple example . . .

Security analyst rule

Certification rule

Countermeasures rule

Robustness (EAL) rule

Expert Knowledge Rule Base

Threats  Policies  Assumptions
Objectives  SFRs  SARs

Rance DeLong and John Rushby
Document Generation

- Current document instantiated from foundation classes of document components and conventions
- Generate / store as XML (e.g.) for flexibility, portability, existing parsers
- Document generators -- semantics-augmented grammars define structure and content requirements
- Workflow rules guide overall process and subprocesses
- Converter to conventional document formats from the current document factbase for further massaging / presentation
GUI Visualization

- Relations
- Statistics
- Guidance interaction
- Rendering of document sections
- Editing the model
- Entering / editing narrative
- Review markup
- Traceability

Collaboration

- Keep central repository of expert knowledge
- No distribution or update headaches
- Seamless way to provide feedback in a semantically rich way
- Medium for formal “buyer-seller” contracts
- Community of authors, reviewers, developers, evaluators, integrators, certifiers
Interaction: Queries (1/2)

- Interrogate the PP “document”
  - “Show all the threats this SFR has been adopted to counter”
  - “Show all the SFRs and SARs that have been specified to achieve this security objective”
  - “Show the tree of objectives and requirements arising from this policy”
  - “What expert knowledge applies to the current document?”

FINE