Common Criteria: Module I

Introduction and General Model

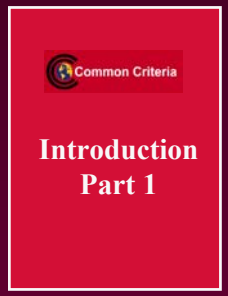
Scope of Common Criteria

- Specification of security properties of IT systems and products that address
  - unauthorized disclosure (confidentiality, privacy)
  - unauthorized modification (integrity)
  - loss of use (availability)
- Basis for the comparison of results of independent evaluations
- Applicable to IT security countermeasures implemented in hardware, software, and firmware
  - independent of technology
  - in user-defined combinations

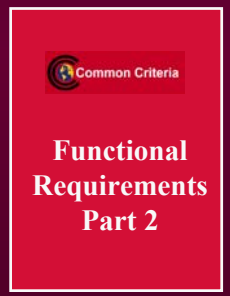
Outside Scope of Common Criteria

- Human and physical security countermeasure implementations
- CC Application
  - ✓ administrative, legal, procedural
  - ✓ certification & accreditation processes
  - ✓ mutual recognition arrangements
- Evaluation methodology
  - ✓ Common Evaluation Methodology for Information Technology Security Evaluation (CEM)
- Cryptographic *algorithm* definition

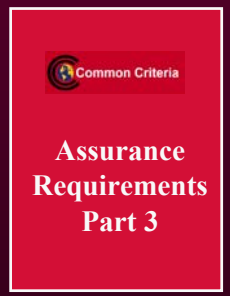
Common Criteria Sections



- Part 1: Introduction and General Model



- Part 2: Security Functional Requirements and Annexes



- Part 3: Security Assurance Requirements

Common Criteria Part 1
(Introduction & General Model)

- Scope, Glossary and Overview
- Security Context and CC Approach
- Security Concepts, Environment and Objectives
- Evaluation Results
- Appendix A: History
- Appendices B: Protection Profile Specification
- Appendices C: Security Target Specification

Common Criteria Terminology

- **Target of Evaluation (TOE)**

  An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

- **Protection Profile (PP)**

  An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.

- **Security Target (ST)**

  A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

- **TOE Security Functions (TSF)**

  A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TOE security policy (TSP).

Common Criteria Terminology

- **Threats**

  Any circumstance or event with the potential to cause harm to a system in the form of destruction, disclosure, modification of data, and /or denial of service.

- **Organizational Security Policy**

  A set of rules, procedures, practices, and guidelines imposed by an organization upon its operations and to which the TOE may have to comply.

- **Secure Usage Assumption**

  Describes the security aspects of the environment in which the TOE will be used or is intended to be used.

- **Security Objective**

  Reflects the intent to counter identified threats and/or address any identified organizational security policies and/or assumptions.

General Principle

- ✓ *ALL TOE security requirements ultimately arise from consideration of the purpose and context of the TOE.*

  This definition requires the PP or ST writer to define a *security environment* which leads to a statement of *security objectives*.

Protection Profiles

- Answers the question:
  What do I need in a security solution?
- Implementation independent
- Multiple implementations may satisfy PP requirements
- Authors can be both consumers and producers of IT products and systems

Security Targets

- Answers the question:
  What do you provide in a security solution?
- Implementation dependent/specific
- Authors can be product vendors, product developers, or product integrators

Protection Profiles and Security Targets

- PP makes a statement of implementation independent security needs
  - ***a generic operating system with discretionary access controls, audit, and identification and authentication***
- ST defines the implementation dependent capabilities of a *specific* product, e.g.
  - ***Microsoft NT 4.0.0.2 (TOE)***
  - ***Sun OS 4.7.4 (TOE)***

PP/ST Comparison

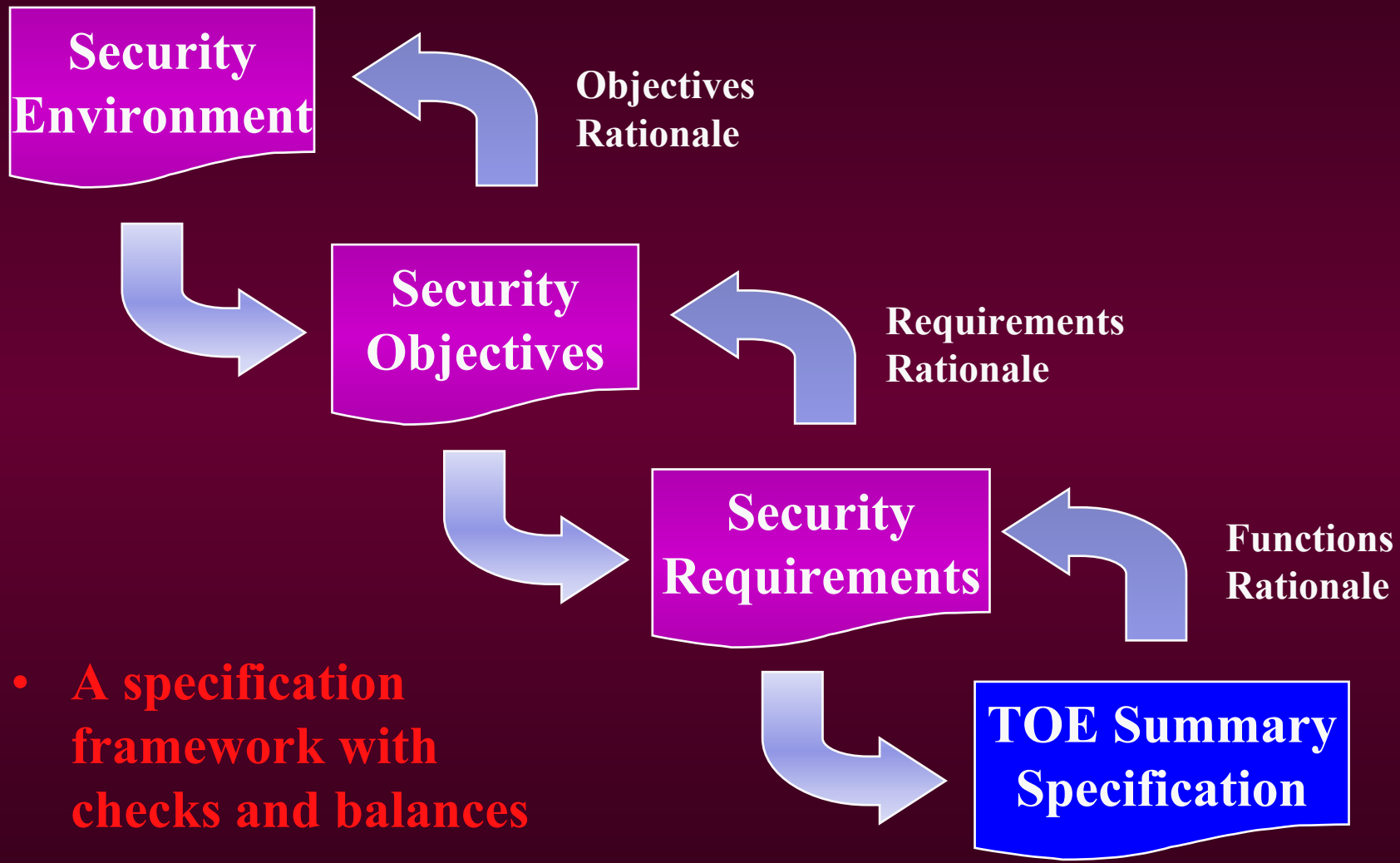
Protection Profile

- Identification
- Overview
- TOE Description
- Security Environment
- Security Objectives
- Security Requirements
- Rationale

Security Target

- Identification
- Overview
- TOE Description
- Security Environment
- Security Objectives
- Security Requirements
- Rationale
- TOE Summary Specification
- CC Conformance Claim
- PP Claims

PP/ST Specification Framework



- A specification framework with checks and balances

Establishing a Security Environment

Things to Consider---

- ✓ TOE physical environment
- ✓ Assets/Resources requiring protection
- ✓ TOE purpose

TOE Security Environment

- Secure Usage Assumptions
  - ✓ *The non-IT security aspects of the environment in which the TOE will be used or is intended to be used.*
- Threats
  - ✓ *The ability to exploit a vulnerability by a threat agent.*
- Organizational Security Policies
  - ✓ *A set of rules, procedures, practices, or guidelines imposed by an organization upon its operations.*

Secure Usage Assumptions

- Describes the security aspects of the environment in which the TOE will be used or is intended to be used
- Information about intended usage and the environment---
  - ✓ intended application, potential asset value, and usage limitations
  - ✓ physical issues, connectivity issues, and personnel issues
  - ✓ must not impose requirements on the TOE or on its IT environment
  - ✓ generate objectives for the (non-IT) environment

Threat

- The ability of a *threat agent* to mount *an attack* on an *asset*, and the *result* of that attack
- Threats provide a basis for statement of countermeasures
- A well-written threat statement addresses
  - ✓ Threat Agent and/or Attacker
  - ✓ The Attack
  - ✓ Assets
  - ✓ Results

Security Policies

- Organizational Security Policy:

  A set of rules, procedures, practices, and guidelines imposed by an organization upon its operations and to which the TOE may have to comply.

- Organizationally-Imposed Requirements
  - ***Passwords Shall Be 8 Characters***
  - ***Cryptography Shall Be Used for Intra-Node Communication***

Environment Examples

- **A.Physical_Protection**

The TOE is installed in a restricted and controlled access area sufficient to prevent unauthorized physical access to the TOE.

- **T.Intercept**

An non-administrative user obtains unauthorized access to controlled information by intercepting information transmitted to/from the TOE.

- **P.Accountability**

The authorized users of the TOE shall be held accountable for their actions within the TOE.

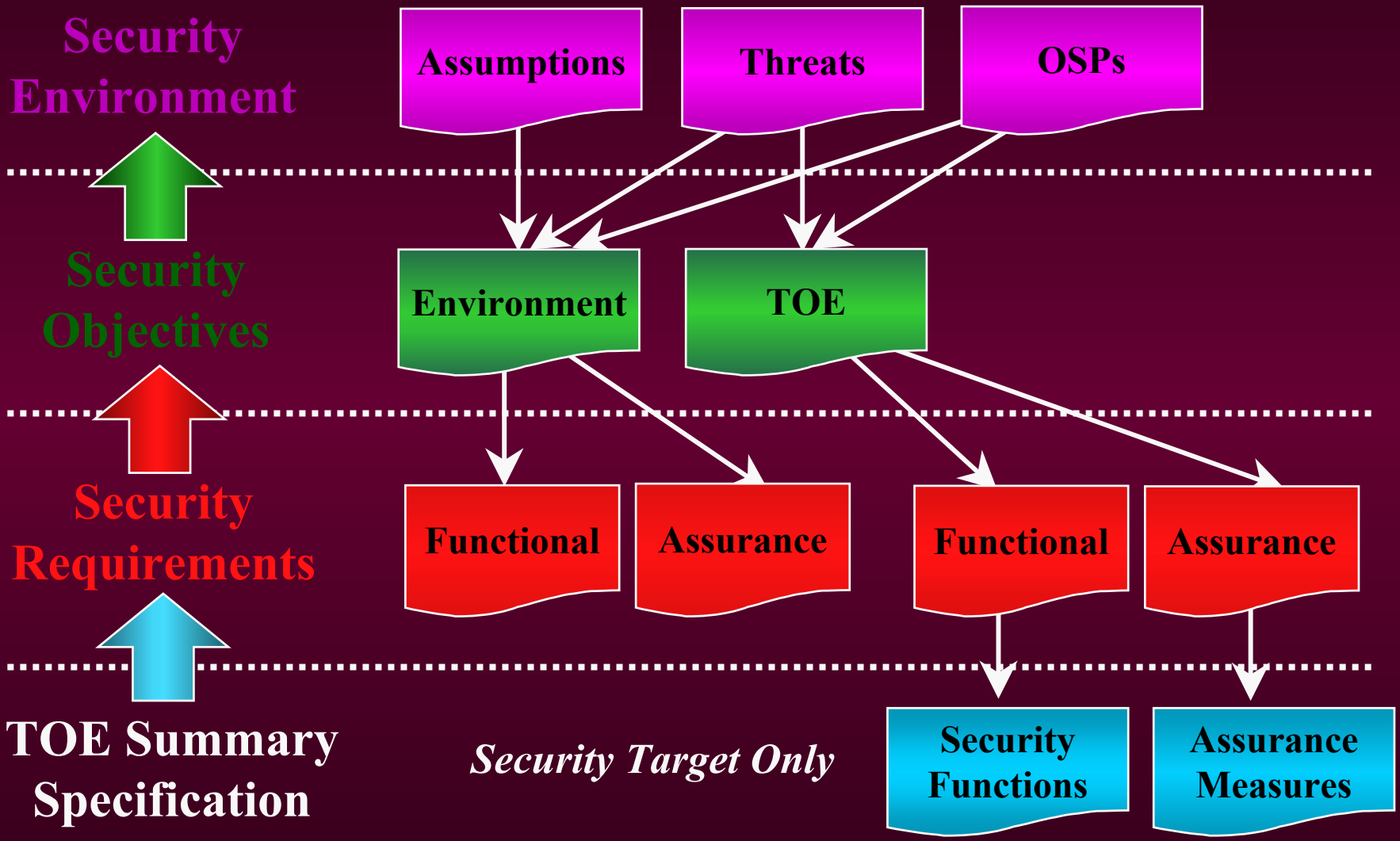
Security Objectives

- Establish the basis for the selection of security requirements (functional & assurance)
- Based completely upon the statement of the security environment
- Objectives describe
  - ✓ Support for assumptions
  - ✓ Mitigation of threats (eliminate, minimize, monitor)
  - ✓ Enforcement organizational security policy

Types of Security Objectives

- Security objectives for the TOE
  - ✓ *Implemented by security requirements allocated to the TOE*
- Security objectives for the environment
  - ✓ *Implemented by security requirements allocated to the IT systems that interact with the TOE*
  - ✓ *Implemented by personnel and procedural means*
  - ✓ *Outside the scope of the CC*

PP/ST Framework



Crafting PPs / STs

- "Top Down" approach
  - Usually PPs
  - Start with environment
  - Derive Objectives
  - Select Requirements
- "Technology Specific" approach
  - Usually PPs
  - Survey products in technology (requirements)
  - Identify function in environment
  - Complete specification
- "Product" approach
  - Usually STs
  - Define what product does (functional requirements)
  - Define existing documentation/ assurance (assurance requirements)
  - "Back in" environment

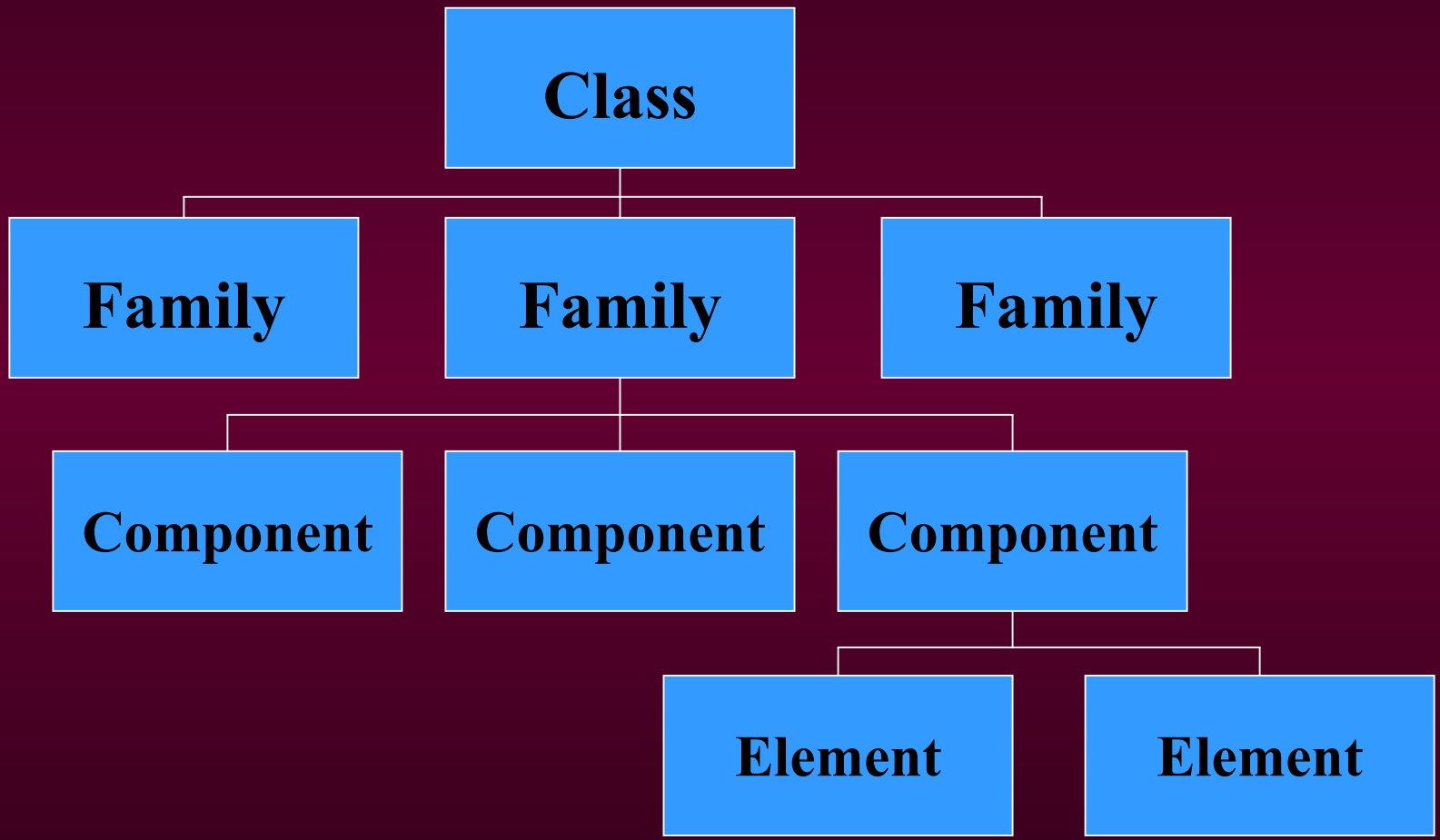
Module II

Security Functional Requirements

Security Functional Requirements

Levied upon functions of the TOE that support IT security; their behavior can generally be observed.

Hierarchy of Requirements
(Functional)



Definitions

- Class - for organizational purposes; all members share a common intent but differ in coverage of security objectives.
- Family - for organizational purposes; all members share security objectives but differ in rigor or emphasis
- Component - describes an actual set of security requirements; smallest selectable set
- Element - members of a component; cannot be selected individually; explicit shall statements

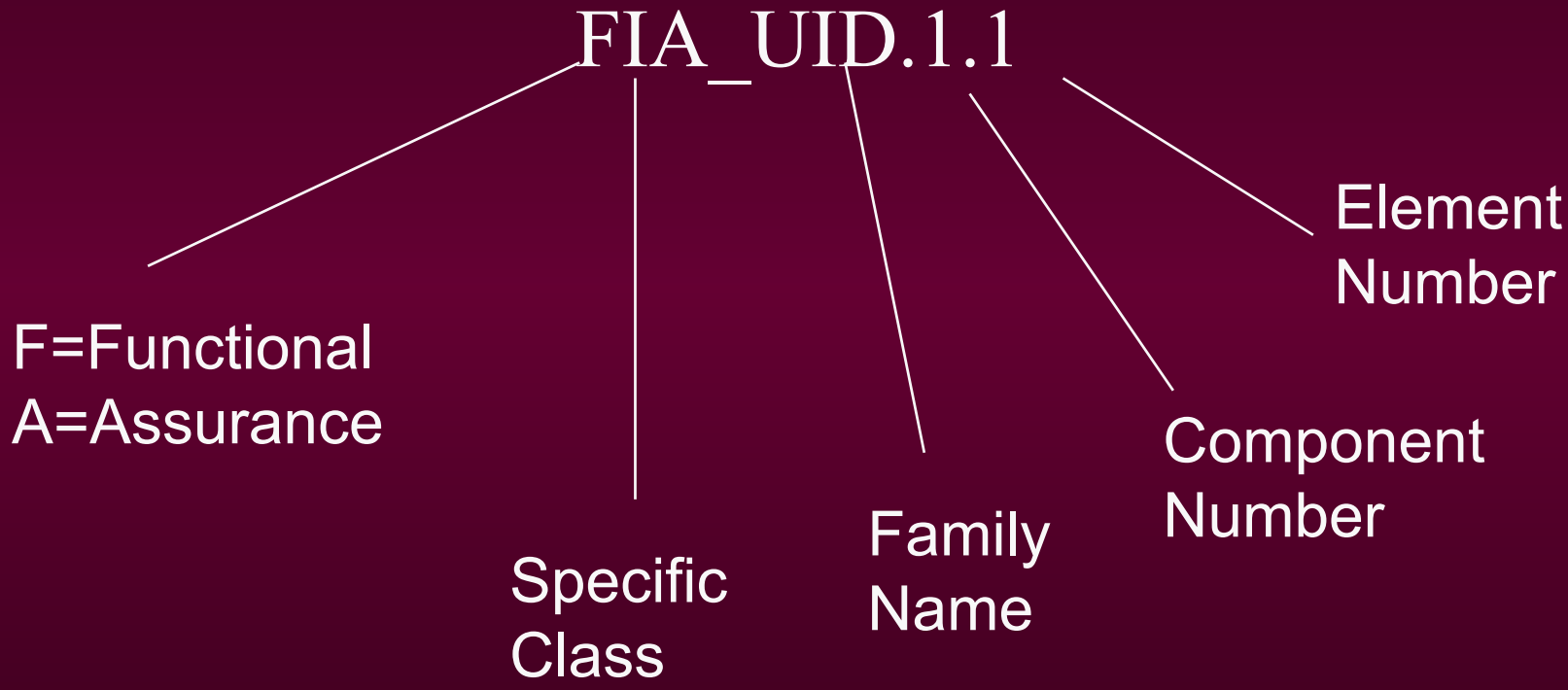
Security Functional Classes

- ✓ Security Audit (FAU)
- ✓ Communications (FCO)
- ✓ Cryptographic Support (FCS)
- ✓ User Data Protection (FDP)
- ✓ Identification & Authentication (FIA)
- ✓ Security Management (FMT)

Security Functional Classes

- ✓ Privacy (FPR)
- ✓ Protection of the Trusted Security Functions (FPT)
- ✓ Resource Utilization (FRU)
- ✓ TOE Access (FTA)
- ✓ Trusted Path (FTP)

Interpreting Functional Requirement Names



Functional Family Structure

FIA_UID User Identification

Family behavior

This family defines the conditions under which users shall be required to identify themselves before performing any other actions that are to be mediated by the TSF and which require user identification.

Component leveling

FIA_UID.1 Timing of identification, allows users to perform certain actions before being identified by the TSF.

FIA_UID.2 User identification before any action, require that users identify themselves before any action will be allowed by the TSF.

Functional Family Structure

Management: FIA_UID.1

The following actions should be considered for the management functions in FMT:

a) the management of the user identities;

b) if an authorized administrator can change the actions allowed before identification, the managing of the action lists.

Management: FIA_UID.2

The following actions should be considered for the management functions in FMT:

a) the management of the user identities.

Audit: FIA_UID.1, FIA_UID.2

The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

a) Minimal: Unsuccessful use of the user identification mechanism, including the user identity provided.

b) Basic: All use of the user identification mechanism, including the user identity provided.

Functional Family Structure

FIA_UID.1 **Timing of Identification**

Hierarchical to: no other components.

FIA_UID.1.1 **The TSF shall allow [assignment: [*list of TSF-mediated actions*] on behalf of the user to be performed before the user is identified.**

FIA_UID.1.2 **The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.**

Dependencies: **No dependencies**

Functional Family Structure

FIA_UID.2 **User Identification before any action**

Hierarchical to: FIA.UID.1

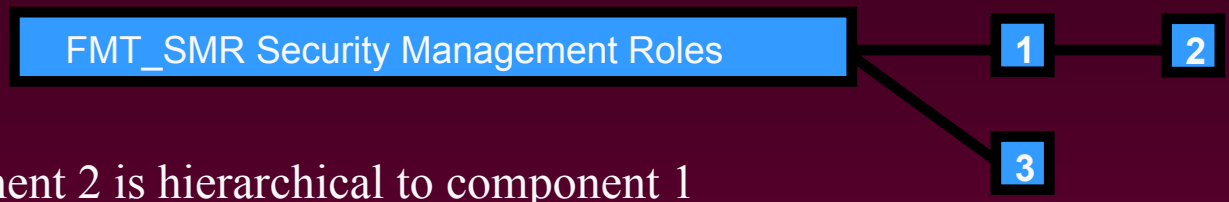
FIA_UID.2.1 **The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.**

Dependencies: **No dependencies**

Component Hierarchy

- Each family contains one or more components
- The component leveling diagram depicts the relationship between components in a family
  - no relationship, or
  - a hierarchical relationship
- A hierarchical component
  - satisfies any dependency on the component it is hierarchical to
  - may provide more security or more functionality than a component it is hierarchical to
- Hierarchical components are not selected together within the same context of the PP/ST

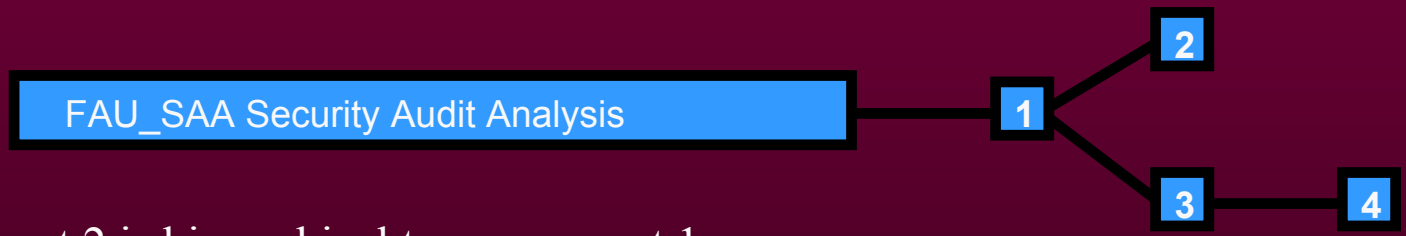
Component Hierarchy Examples



Component 2 is hierarchical to component 1

Component 3 is not hierarchically related to either component 1 or 2

Legal component selections are: Component 1, Component 2, Component 3, Components 1 and 3, Components 2 and 3



Component 2 is hierarchical to component 1

Component 3 is hierarchical to component 1

Component 4 is hierarchical to component 3

Legal component selections are: Component 1, Component 2, Component 3, Component 4, Components 2 and 3, Components 2 and 4

Class FAU: Security Audit

- Common Intent: The six families in this class are concerned with ...
  - recognizing and responding to (FAU_SAA, FAU_ARP)
  - recording (FAU_GEN, FAU_SEL)
  - storing and protecting (FAU_STG)
  - review and analysis of (FAU_SAR)

... security-relevant events and activities.

Class FAU: Security Audit
(An Example)

NEED:

A record of certain actions taken by users such that an administrator can determine when the action occurred, who did it, whether it succeeded or failed.

TO SATISFY:

- ✓ FAU_GEN.1 Audit Data Generation
- ✓ FAU_GEN.2 User Identity Association

Class FCO: Communication

- Common Intent: The two families in this class are concerned with ...
  - proof of origin (FCO_NRO)
  - proof of receipt (FCO_NRR)

... of transmitted information.

Class FCO: Communication
(An Example)

NEED:

The recipient of all email messages must be able to verify the identity of the sender.

TO SATISFY:

- ✓ FCO_NRO.1 Selective Proof of Origin
- ✓ FCO_NRO.2 Enforced Proof of Origin (more functionality)

Class FCS: Cryptographic Support

- Common Intent: The two families in this class are concerned with ...
  - Generation, distribution, access, and destruction (FCS_CKM)
  - operational use (FCS_COP)

... of cryptographic keys.

Class FCS: Cryptographic Support
(An Example)

NEED:

An administrator must generate and distribute cryptographic keys according to the appropriate algorithms and distribution, respectively.

TO SATISFY:

- ✓ FCS_CKM.1 Cryptographic Key Generation
- ✓ FCS_CKM.2 Cryptographic Key Distribution

Class FDP: User Data Protection

- Common Intent: The thirteen families in this class are concerned with ...
  - security function policies (FDP_ACC, FDP_IFC)
  - forms of user data protection (FDP_ACF, FDP_IFF, FDP_ITT, FDP_RIP, FDP_ROL, FDP_SDI)
  - import/export (FDP_DAU, FDP_ETC, FDP_ITC)
  - inter-TSF communications (FDP_UCT, FDP_UIT)

... for data protection.

Class FDP: User Data Protection
(An Example)

NEED:

When a user data file is deleted its contents must be inaccessible and when a new one is created it should contain no previous information.

TO SATISFY:

- ✓ FDP_RIP.2 Full Residual Information Protection

Class FIA: Identification & Authentication

- Common Intent: The six families in this class are concerned with ...
  - identification (FIA_UID)
  - authentication (FIA_UAU, FIA_SOS, FIA_AFL)
  - attributes (FIA_ATD, FIA_USB)

... of a user.

Class FIA: Identification & Authentication
(An Example)

NEED:

An individual may only attempt to log into the system 3 times. After that, if the attempts are not successful, the individual's account shall be locked until unlocked by an administrator.

TO SATISFY:

- ✓ FIA_AFL.1 Basic Authentication Handling

Class FMT: Security Management

- Common Intent: The six families in this class are concerned with ...
  - management of TSF data (FMT_MTD)
  - management of security attributes (FMT_MSA, FMT_REV, FMT_SAE)
  - management of security functions (FMT_MOF)
  - security roles (FMT_SMR)

... of the TOE.

Class FMT: Security Management
(An Example)

NEED:

Our organization has a security officer responsible for new users and I&A functions; and an audit administrator responsible for the audit mechanism.

TO SATISFY:

- ✓ FMT_SMR.1 Security Management Roles
- ✓ FMT_MTD.1 Management of TSF Data

Class FPR: Privacy

- Common Intent: The four families in this class are concerned with protection against ...
  - discovery and misuse (FPR_ANO, FPR_PSE, FPR_UNL, FPR_UNO)

... of an individual's identity by others.

Class FPR: Privacy
(An Example)

NEED:

A web page's content and questionnaire deal with a sensitive public health issue. It is important that respondents be assured of complete unobservability when reading the data and filling out of the form. There is also no reason for even an administrator to be capable of identifying individuals who choose to respond. Without such assurance, people will be reluctant to respond and the sponsoring authority will not get accurate data.

TO SATISFY:

- ✓ FPR_UNO.1 Unobservability

Class FPT: Protection of the Trusted Security Functions

- The sixteen families in this class address ...
  - reference mediation and domain separation (FPT_RVM, FPT_SEP)
  - testing (FPT_AMT, FPT_TSF)
  - physical/anti-tamper protection (FPT_PHP)
  - secure TSF data transfer (FPT_ITA, FPT_ITC, FPT_ITI, FPT_ITT, FPT_RPL, FPT_TDC, FPT_TRC)
  - failure and recovery (FPT_RCV, FPT_FLS)
  - state and timing (FPT_SSP, FPT_STM)

... of the TSF mechanisms and data.

Class FPT: Protection of the Trusted Security Functions
(An Example)

NEED:

An authorized administrator must be able to verify that the executables that implement the security functions have not been modified by malicious individuals or code.

TO SATISFY:

- ✓ FPT_TST.1 TSF Self Test

Class FRU: Resource Utilization

- Common Intent: The three families in this class are concerned with ...
  - availability (FRU_FLT)
  - allocation (FRU_PRS, FRU_RSA)

... of resources.

Class FRU: Resource Utilization
(An Example)

NEED:

A denial of service attack by a user consuming all available disk space must be prevented.

TO SATISFY:

- ✓ FRU_RSA.1 Maximum Quotas

Class FTA: TOE Access

- Common Intent: The six families in this class are concerned with ...
  - attributes (FTA_LSA, FTA_TAB, FTA_TAH)
  - establishment and control (FTA_MCS, FTA_SSL, FTA_TSE)

... of a user session.

Class FTA: TOE Access
(An Example)

NEED:

Whenever a user session remains idle for a specified period of time, the session shall be automatically locked by the system. Also, individuals shall have the ability to lock their own sessions.

TO SATISFY:

- ✓ FTA_SSL.1 TSF-Initiated Locking
- ✓ FTA_SSL.2 User-Initiated Locking

Class FTP: Trusted Path/Channels

- Common Intent: The two families in this class are concerned with ...
  - trusted communication paths (FTP_TRP)
  - trusted communication channels (FTP_ITC)

... between users and the TSF; and between the TSF and other trusted IT products, respectively.

Class FTP: Trusted Path/Channel
(An Example)

NEED:

There must be a means by which remote administrators can verify that they are communicating with the TSF.

TO SATISFY:

- ✓ FTP_TRP.1 Trusted Path

Requirements Rationale

- Threats/OSPs (through security objectives) drive functional requirement selection
- Rationale must demonstrate that the functional requirements are ***suitable to meet and traceable to*** the security objectives
- The rationale must demonstrate:
  - ✓ why the choice of security requirements meets an objective
  - ✓ functional & assurance requirements are not contradictory and are complete
  - ✓ strength of function (SOF) claims are consistent with the security objectives

Operations on Requirements
(Functional)

- Types of operations
  - ✓ assignment
  - ✓ selection
  - ✓ refinement
  - ✓ iteration
- Functional requirements have placeholders indicating where assignment and selection operations are allowed
- Refinement and iteration may be performed on any functional requirement

Assignment Operations

- Specification of a parameter filled in when component is used
- “Fill in the Blank” operation
- Allows PP/ST writer to provide information relating to application of the requirement
- The PP writer may defer completing assignments, but the ST writer must complete all assignments

Assignment Operation
(An Example)

As Written in the Common Criteria:

- **FMT_SMR.1.1** The TSF shall maintain the roles: [assignment: *the authorized identified roles*].

After Assignment Operation:

- **FMT_SMR.1.1** The TSF shall maintain the roles: [*auditor, security administrator, operator*].

Selection Operations

- Specification of elements selected from a list given in the component
- “Multiple Choice” operation
- Allows PP/ST writer to select from a provided list of choices
- The PP writer may defer completing selections, but the ST writer must complete all selections

Selection Operation
(An Example)

As Written in the Common Criteria:

- **FTA_TAH.1.1** Upon successful session establishment, the TSF shall display the [selection: *date, time, method, location*] of the last successful session establishment to the user.

After Selection Operation:

- **FTA_TAH.1.1** Upon successful session establishment, the TSF shall display the [*date, time, and location*] of the last successful session establishment to the user

Selection and Assignment
(An Example)

As Written in the Common Criteria:

- **FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear,* [assignment: *other operations*]] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].

After Operations:

- **FMT_MTD.1.1** The TSF shall restrict the ability to [*delete*, [*and create*]] the [*user authentication database*] to [*the security administrator*].

Refinement Operations

- A mechanism to tailor a requirement by specifying additional detail in order to meet a security objective
- Can be performed on any functional component
- Rules for refinement:
  - ✓ the refinement shall only restrict the set of possible acceptable functions used to implement the requirement
  - ✓ the refinement may not levy completely new requirements
  - ✓ the refinement may not increase the list of dependencies of the requirement being refined
  - ✓ the refinement may provide an elaboration or interpretation
  - ✓ the refinement may not eliminate the requirement

Refinement Operation
(An Example)

As Written in the Common Criteria:

- **FAU_SAA.1.1** The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

After Refinement Operation:

- **FAU_SAA.1.1** The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP *by notifying the Security Officer immediately*.

Iteration Operations

- Repetitive use of the same component to address different aspects of the requirement being stated (e.g., identification of more than one type of user).
- Can be performed on any functional component

Iteration Operation
(An Example)

As Written in the Common Criteria:

- **FMT_MTD.1.1** The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear*, [assignment: *other operations*]] the [assignment: *list of TSF data*] to [assignment: *the authorized identified roles*].

After Iteration Operation:

- **FMT_MTD.1.1(1)** The TSF shall restrict the ability to [*modify*] the [*enrolled images db*] to [*the security administrator*].
- **FMT_MTD.1.1(2)** The TSF shall restrict the ability to [*backup/restore*] the [*enrolled images db*] to [*the operator*].

Dependencies
(Functional Components)

- Some requirement components are not self sufficient
- Some functional requirement components have functional and assurance dependencies
- Some dependencies may be eliminated with sufficient rationale

Common Criteria Part 2: Annexes

- Annex A:

  Security Functional Requirements Application Notes
  - Dependency Table
- Annexes B - M:

  Similar to Part 2 but more informative
  - ✓ user notes
  - ✓ evaluator notes
  - ✓ documentation notes

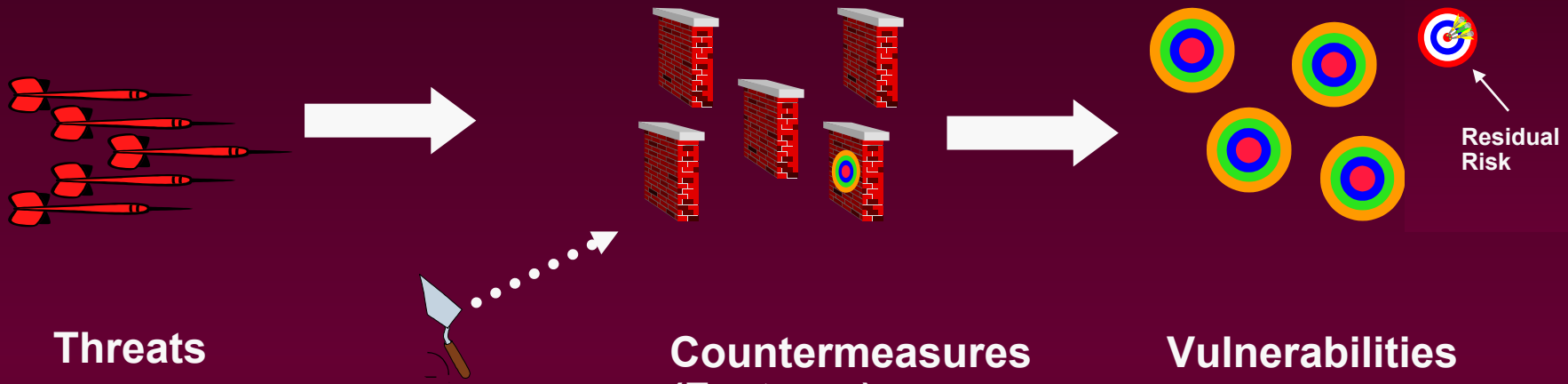
Module III

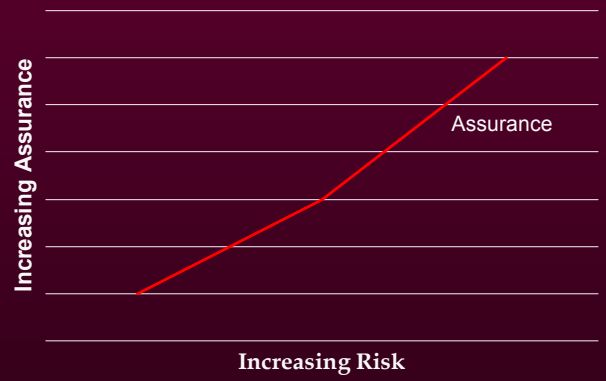
Security Assurance Requirements

What is Assurance?

Assurance is a property of the TOE which gives confidence that the claimed security measures of the TOE are effective and implemented correctly.

Why Do We Care About Assurance?





National Information Assurance Partnership®

How Do We Gain Assurance?

- Analysis of the correspondence between TOE design representations
- Analysis of the TOE design representations against the requirements

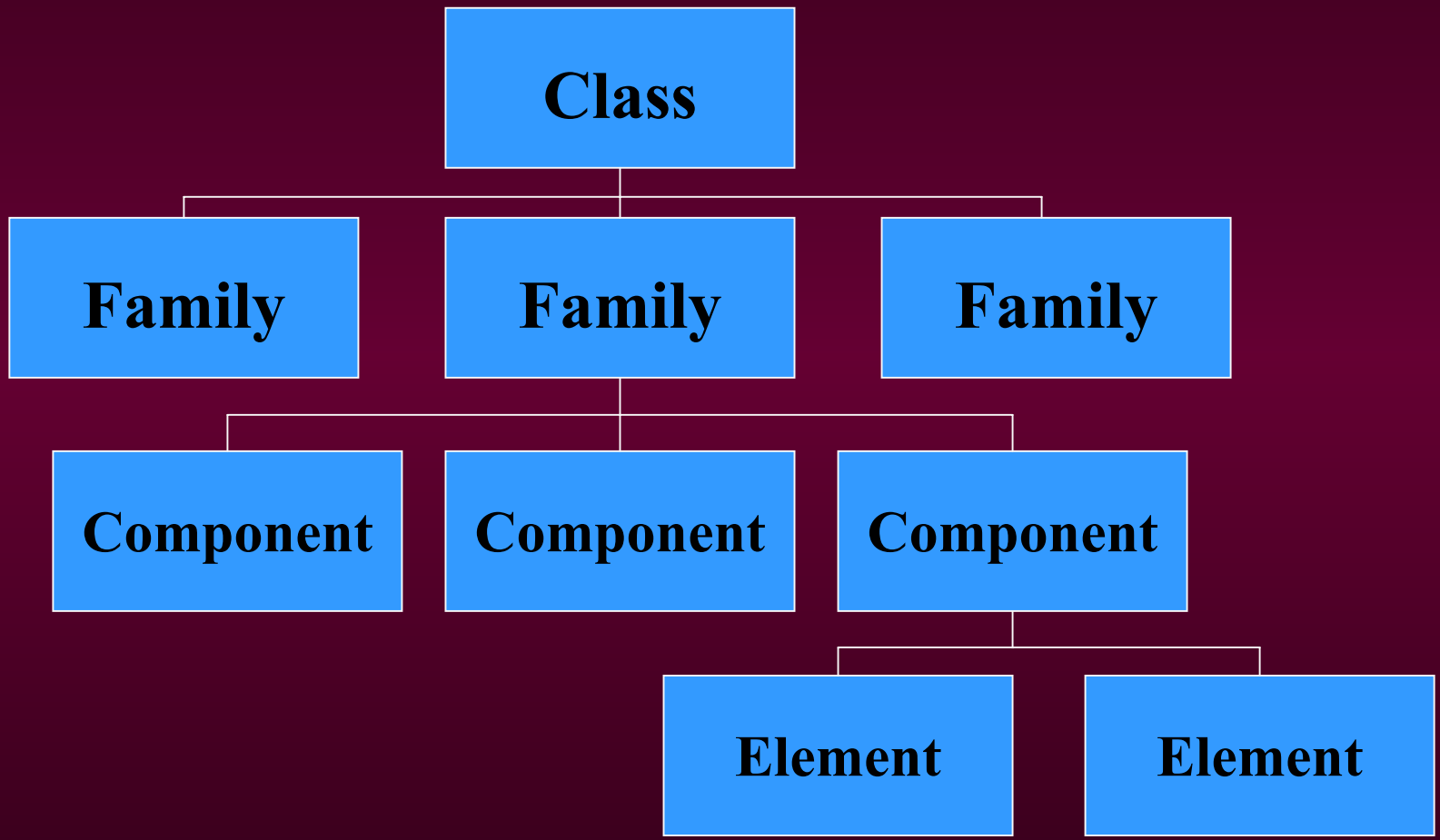
- Analysis of functional tests coverage, and results
- Independent functional testing
- Penetration testing
- Verification of mathematical proofs
- Analysis of guidance documents
- Analysis of processes and procedures
- Checking that processes and procedures are being applied

Evaluation Assurance Scale

Greater Evaluation Effort (Scope, Depth, Rigor)

Greater Assurance

Hierarchy of Requirements
(Assurance)



Definitions

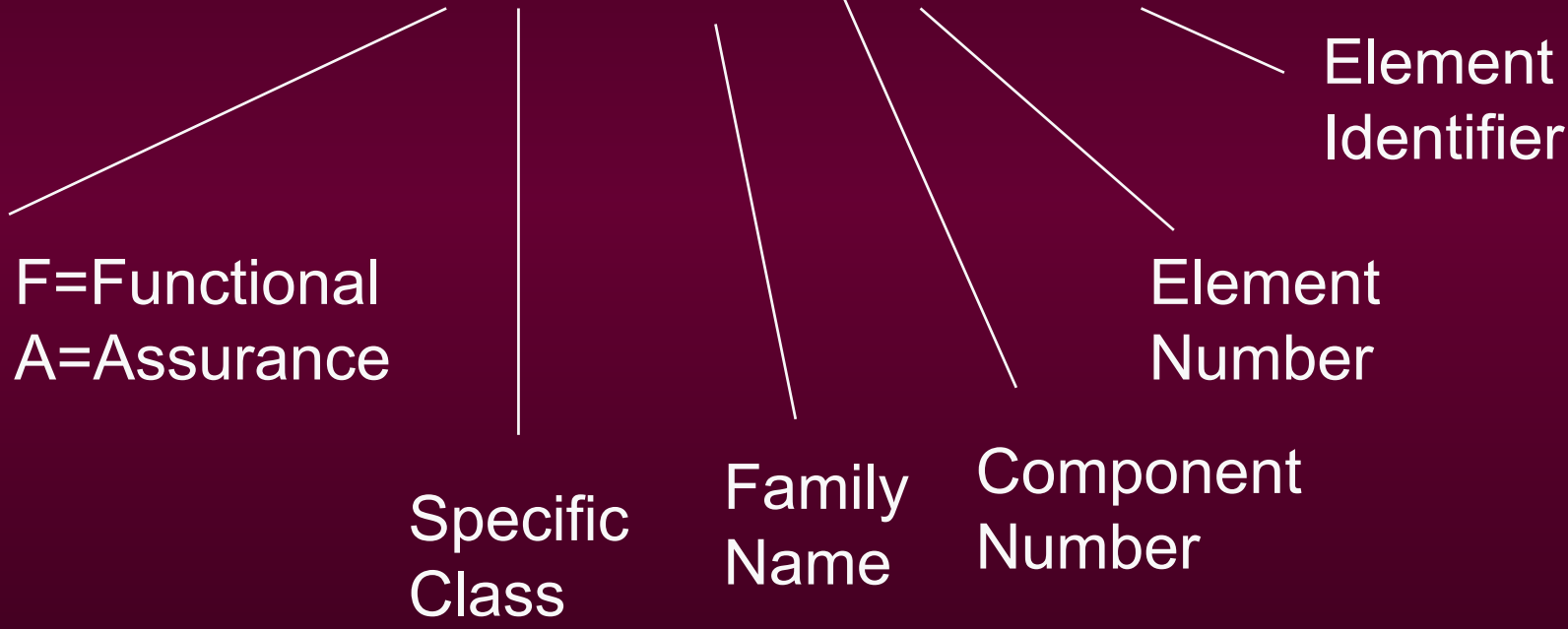
- Class - for organizational purposes; all members share a common intent but differ in coverage of security objectives.
- Family - for organizational purposes; all members share security objectives but differ in rigor or emphasis
- Component - describes an actual set of security requirements; smallest selectable set
- Element - members of a component; cannot be selected individually; explicit shall statements

Security Assurance Classes

- ✓ Configuration Management (ACM)
- ✓ Delivery and operation (ADO)
- ✓ Development (ADV)
- ✓ Guidance documents (AGD)
- ✓ Life Cycle Support (ALC)
- ✓ Tests (ATE)
- ✓ Vulnerability assessment (AVA)
- ✓ Evaluation Criteria (APE, ASE)
- ✓ Assurance Maintenance (AMA)

Interpreting Assurance Requirement Names

ADV_LLD.3.1(D,C,E)



Class ACM: Configuration Management

- Common Intent: The three families in this class are concerned with ...
  - protecting the integrity (ACM_SCP)
  - tracking/restricting the modification (ACM_AUT, ACM_CAP)

... of configuration items.

Class ADO: Delivery and Operation

- Common Intent: The two families in this class are concerned with ...
  - delivery (ADO_DEL)
  - installation, generation, start-up (ADO_IGS)

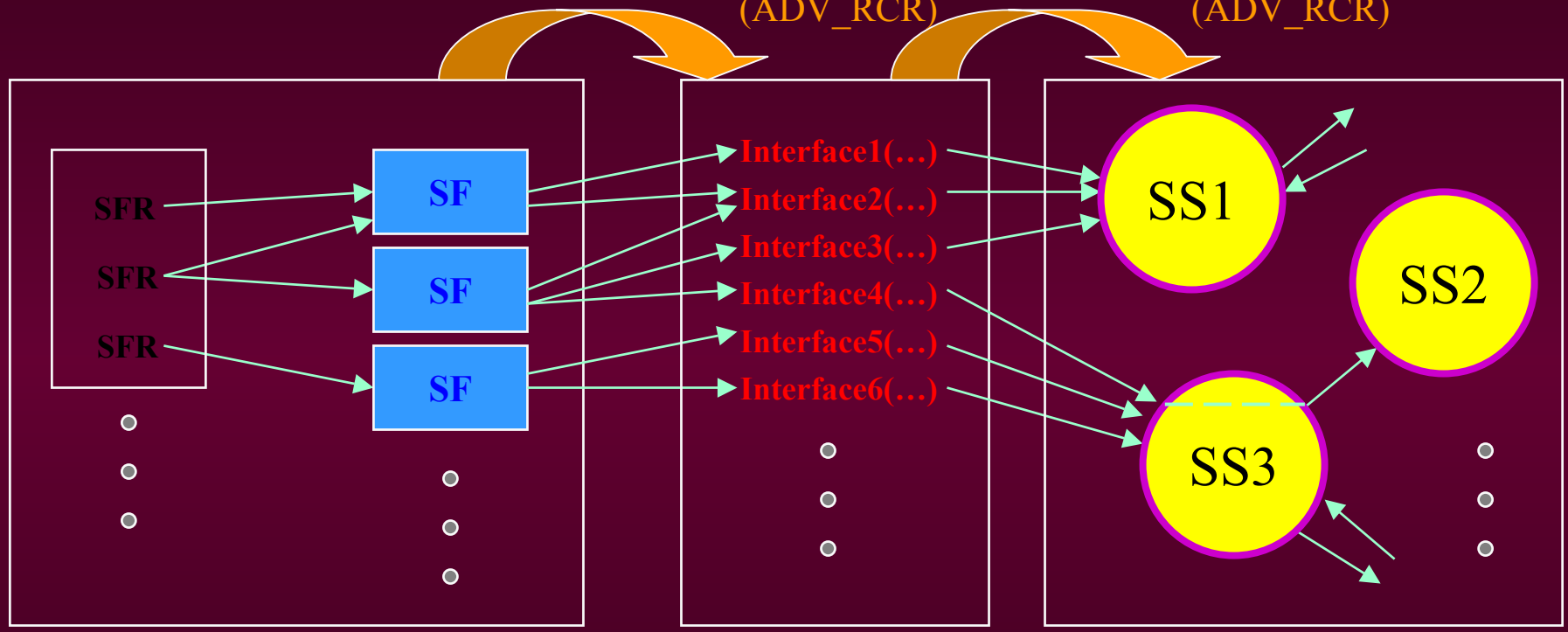
... of the TOE.

Class ADV: Development

- Common Intent: The seven families in this class are concerned with ...
  - levels of abstraction (ADV_FSP, ADV_HLD, ADV_IMP, ADV_LLD)
  - correspondence mapping of representations (ADV_RCR)
  - internal structure (ADV_INT)
  - policy model (ADV_SPM)

... of the TSF.

ADV Overview

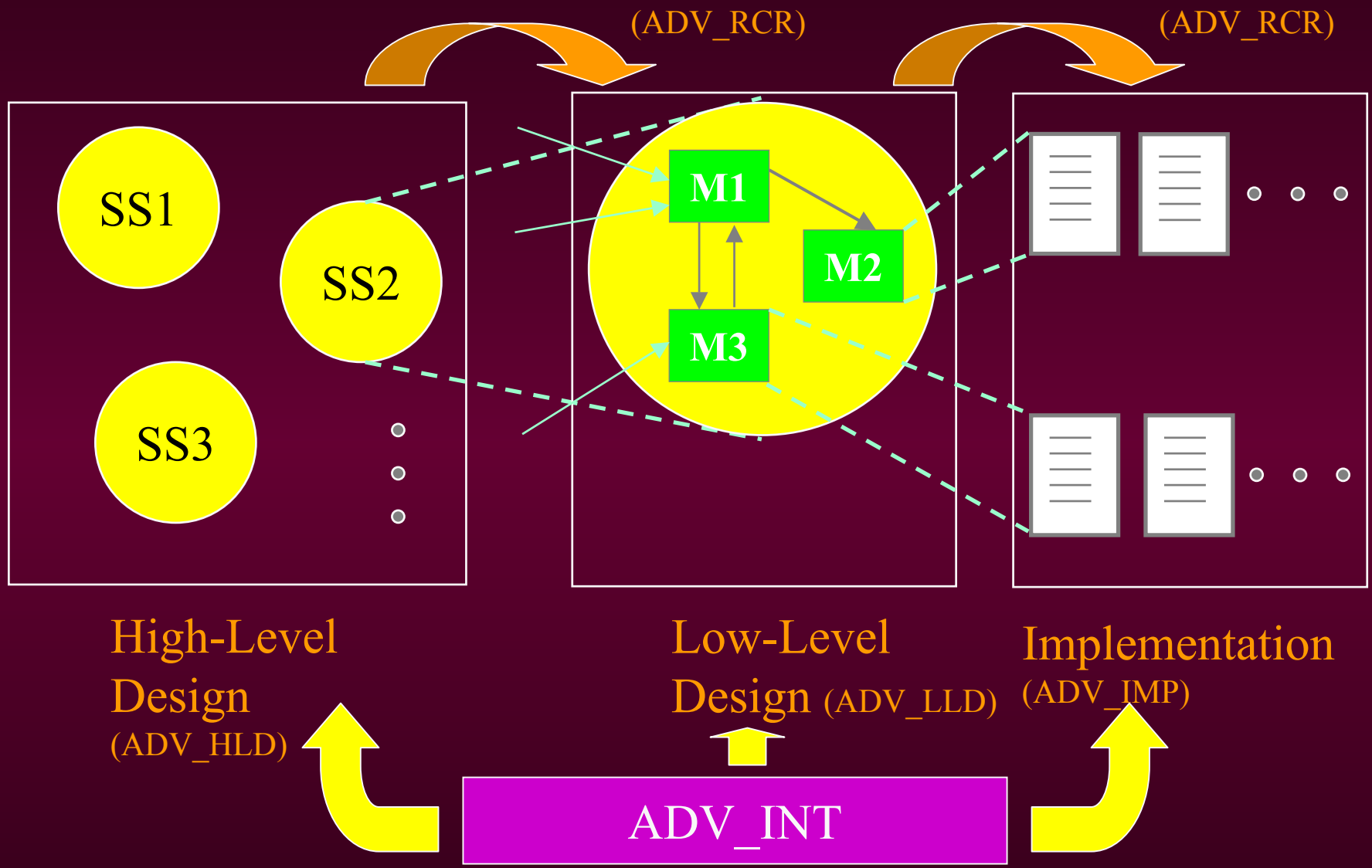


Security Target

Functional Specification
(ADV_FSP)

High-Level Design
(ADV_HLD)

ADV Overview



Class AGD: Guidance Documents

- Common Intent: The two families in this class are concerned with ...
  - user (AGD_USR)
  - administrator (AGD_ADM)

... guidance documentation.

Class ALC: Life Cycle Support

- Common Intent: The four families in this class are concerned with refinement of the TOE during ...
  - development (ALC_DVS, ALC_FLR)
  - maintenance (ALC_LCD, ALC_TAT)

... phases.

Class AMA: Maintenance of Assurance

- Common Intent: The four families in this class are concerned with...
  - maintenance planning & procedures (AMA_AMP, AMA_EVD)
  - maintenance activities (AMA_CAT, AMA_SIA)

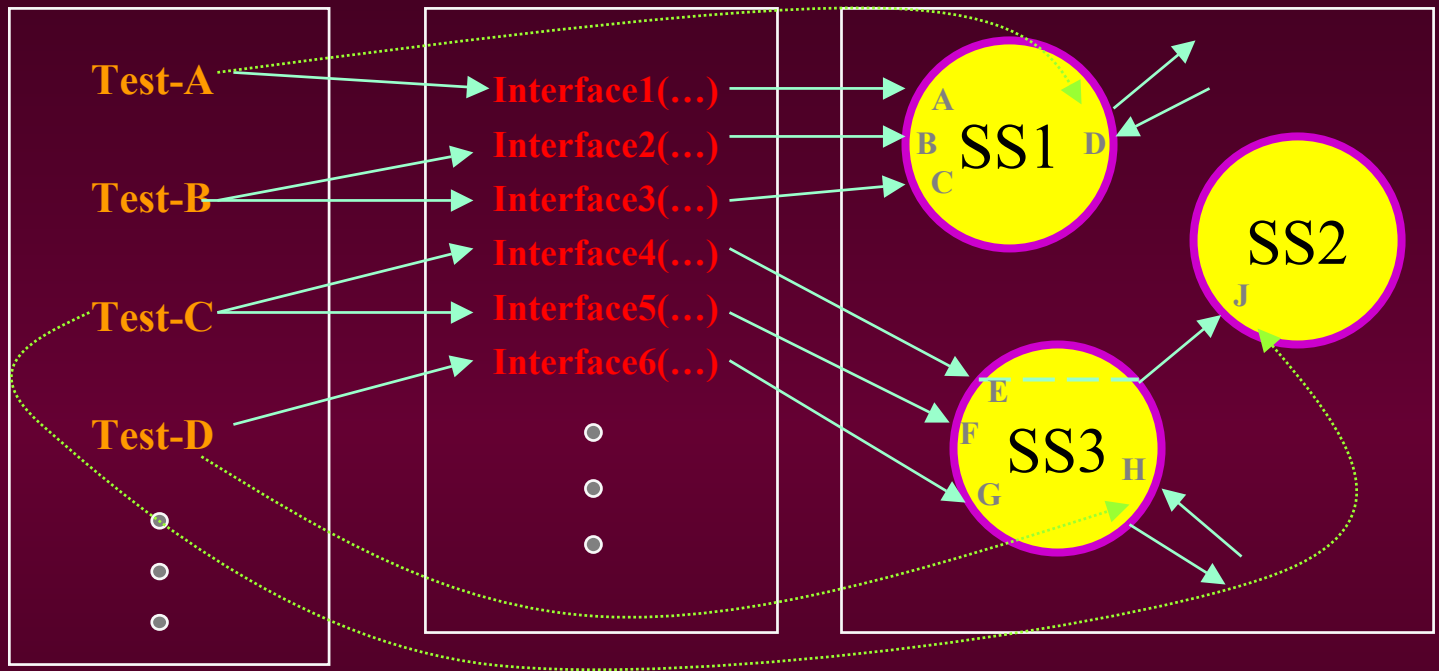
... after a TOE has been evaluated against the CC.

Class ATE: Tests

- Common Intent: The four families in this class are concerned with ...
  - coverage (ATE_COV)
  - depth (ATE_DPT)
  - vendor functional and independent (ATE_FUN)
  - evaluator independent (ATE_IND)

... testing.

ATE_COV, ATE_DPT



Test Procedure Descriptions

Functional Specification

High-Level Design

Class AVA: Vulnerability Assessment

- Common Intent: The four families in this class are concerned with ...
  - exploitable covert channels (AVA_CCA)
  - misuse (AVA_MSU)
  - vulnerabilities and strength (AVA_VLA, AVA_SOF)

... of the TOE.

Class APE: Protection Profile Evaluation

- Common Intent: The six families in this class are concerned with ...
  - complete, consistent, and technically sound (APE_DES, APE_ENV, APE_INT, APE_OBJ, APE_REQ, APE_SRE)

... protection profiles.

Class ASE: Security Target Evaluation

- Common Intent: The eight families in this class are concerned with ...
  - complete, consistent, and technically sound (ASE_DES, ASE_ENV, ASE_INT, ASE_OBJ, ASE_PPC, ASE_REQ, ASE_SRE, ASE_TSS)

... security targets that are suitable for TOE specification.

Dependencies
(Assurance Components)

- Dependencies have same meaning as for functional requirements
- Table A.1 (Part 2: Annexes page 4) identifies all dependencies
  - direct (as stated in the requirement)
  - indirect (as a result of "chasing down" the dependencies)

Operations on Requirements
(Assurance)

- Iteration
- Refinement

Requirements Packages

- Reusable set of *functional* or *assurance* components combined together to satisfy a set of identified security objectives
- In CC Part 3 there are 7 assurance packages called Evaluation Assurance Levels (increasing rigor and formalism from EAL1 to EAL7)
- Packages being specified for levels of robustness
  - Basic and Medium are in draft
  - High is still being defined

Evaluation Assurance Levels (EALs)

- Provide an increasing scale
- This scale balances:
  - ✓ level of assurance obtained
  - ✓ cost/feasibility of acquiring it

Considerations for EAL Selection

- ✓ Value of the assets
- ✓ Risk of the assets being compromised
- ✓ Current state of practice in definition and construction of the TOE
- ✓ Security Environment
- ✓ Development, evaluation, & maintenance costs
- ✓ Resources of adversaries
- ✓ Functional requirement dependencies

EAL1 - Functionally Tested

- Confidence in current operation is required
- No assistance from TOE developer
- Applicable where threat to security is not serious
- Incomplete independent testing against specification and guidance documentation

EAL2: Structurally Tested

- Requires some cooperation of the developer
- Low to moderate of independently assured security
- Adds requirements for configuration list, delivery, high-level design documentation, developer functional testing, vulnerability analysis, more extensive (but still not complete) independent testing

EAL3: Methodically Tested and Checked

- Requires positive security engineering at the design stage without substantial changes in existing practices
- Moderate assurance through investigation of product and development environment controls, and high-level design documentation
- Places additional requirements on testing (now complete), development environment controls and TOE configuration management

EAL4: Methodically Designed, Tested, and Reviewed

- Requires security engineering based on good commercial development practices
- Highest level likely for retrofit of an existing product
- Additional requirements on design, implementation, vulnerability analysis, low level design documentation, development and system automated configuration management, and an informal security policy model

EAL5: Semiformally Designed and Tested

- Higher assurance, risk situations
- Requires rigorous commercial development practices and moderate use of specialist engineering techniques
- Introduces structured implementation of TSF
- Additional requirements on semi-formal functional specification, high-level design, and their correspondence, increased vulnerability testing, full implementation representation, and covert channel analysis

EAL6: Semiformally Verified Design and Tested

- Applicable to a rigorous development environment
- High assurance for high value assets/risk situations
- Additional requirements on analysis, layered TOE design, semi-formal low-level design documentation, complete CM system automation and a structured development environment, and increased vulnerability testing/covert channel analysis

EAL7: Formally Verified Design and Tested

- Maximum assurance for extremely high risk situations
- Generally for experimental application
- Assurance is gained through application of formal methods in the documentation of the functional specification and high-level design
- Additional requirements for complete developer test analysis, complete independent confirmation of the test results, and complete documentation of the structure of the TSF

EAL Augmentation

- The tailoring of an existing Evaluation Assurance Level (EAL)
  - ✓ Specify assurance component(s) in addition to those in an existing EAL
- Allowed augmentation operations
  - ✓ Specify a higher component in the same family
  - ✓ Specify a higher component from another family
  - ✓ Specify new components that are not contained in an EAL
- Disallowed augmentation operation
  - ✓ Removal of components from an EAL definition

U.S. Government Packages

- Based on DoDI 8500.2 and NIST guidance, U.S. Government Protection Profiles are developed according to the following defined packages:
  - U.S. Government Basic Robustness
  - U.S. Government Medium Robustness
  - U.S. Government High Robustness

Basic Robustness

- Basic Robustness provides assurance by an analysis of the TOE security functions using
  - guidance documentation,
  - functional specification,
  - high level design, and
  - interface specification.
- EAL 2 augmented portions require
  - accuracy of system documentation,
  - the tracking and correction of system flaws.

Basic Robustness (cont.)

- Assurance requirements include all components of EAL 2 augmented with
  - ✓ Flaw Reporting Procedures (ALC_FLR.2)
  - ✓ Examination of Guidance (AVA_MSU.1)

- Allow "Partial" TOEs
  - ✓ Software only
  - ✓ Portion of system (e.g., database only)

Medium Robustness

- Medium robustness provides assurance by an analysis of the TOE security functions using
  - architectural design documents,
  - low-level design of the TOE,
  - implementation representation of the entire TSF,
  - complete interface specifications,
  - systematic cryptographic module covert channel,
  - informal TOE security policy model, and
  - modular TOE design.

- Allow only "complete" TOEs (i.e. hardware, operating system, and application software are required).

Medium Robustness (cont)

- Medium robustness includes components of EAL 4 augmented with
  - ✓ Implementation of the TSF (ADV_IMP.2)
  - ✓ Testing: Low-level Design (ATE_DPT.2)
  - ✓ Flaw Reporting Procedures (ALC_FLR.2)
  - ✓ Moderately Resistant (AVA_VLA.3)
  - ✓ Functional Specification (ADV_FSP_(EXP).1
  - ✓ Security-enforcing High-level design (ADV_HLD_(EXP).1)
  - ✓ Security-enforcing Low-level design (ADV_LLD_(EXP).1
  - ✓ Architectural Design with Justification (ADV_ARC_(EXP).1
  - ✓ Modular Decomposition (ADV_INT_(EXP).1)
  - ✓ Systematic Cryptographic Module Covert Channel Analysis (AVA_CCA_(EXP).1)

High Robustness

- High robustness will build upon Medium robustness requirements and are currently being targeted at the EAL 6 level.

- The exact assurance requirements are still being developed. Completion date is TBD.

Contact Information

Department of Defense
ATTN: Jean Schaffer
9800 Savage Rd
Fort Meade, Maryland 20755-6740
Jhschaf@missi.ncsc.mil

Aerospace Corporation
ATTN: Ken Elliott
8840 Stanford Blvd, Suite 4400
Columbia, Maryland 21045
elliott@aero.org

Common Criteria

Ms. Jean Schaffer
NIAP Director
(410) 854-4458

Mr. Ken Elliott
Senior Validator
(410) 312-1405