Common Criteria

Johan Otterström

Sectra Communications AB

SECTRA

Sectra Communications AB

Cryptographic Tokens

Network Encryption

Management Equipment

Personal Devices

Radio

SECTRA

Outline

- Security Evaluation
- Common Criteria
- Development Workflow

Security Evaluation

- Independent verification of security claims
- Determine the appropriateness of security functions and assurance
- Reveal weaknesses

Methods

- Common Criteria
- FIPS 140, Security Requirements for Cryptographic Modules
- National standards and requirements

Why evaluate?

- Buyer:
  - To get assurance of the security in the product
  - Independent statement of the security
- Supplier
  - Legal requirements, legislation, etc.
  - Competitive advantage

Common Criteria

- Internationally recognized standard for evaluating security products
- Evaluation is performed by an independent and certified entity (evaluation facility)
- Product that pass the evaluation gets a certificate
- The certificate is valid for all countries that is part of the Common Criteria community

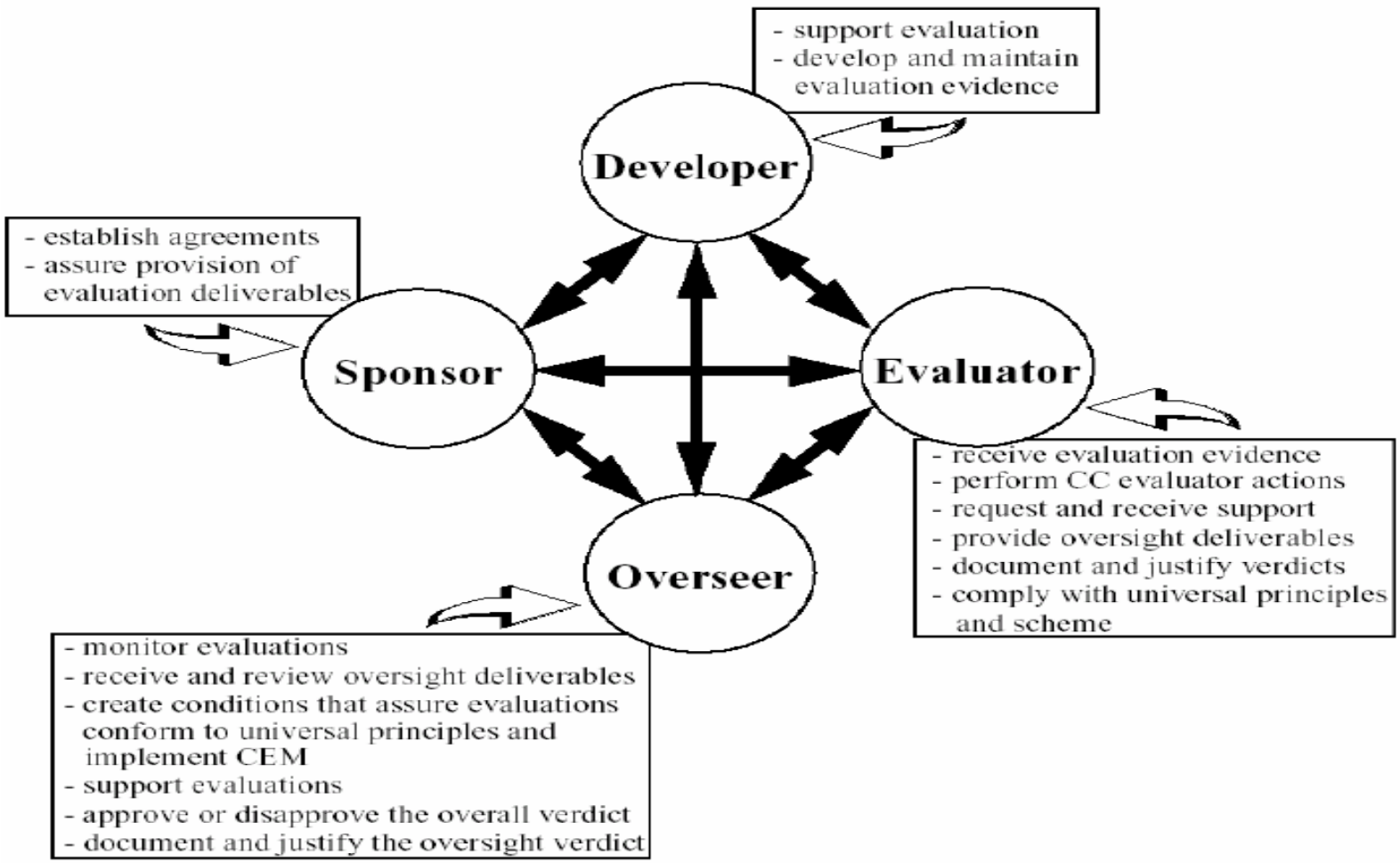
Common Criteria

- Rules for:
  - Security requirements and security function specification
  - The development process
    - Work flow, testing
  - Development environment
    - Configuration management, security
  - User documentation
  - Operational environment
  - Product lifecycle

Common Criteria Documentation

- Common Criteria (ISO/IEC 15408)
  - Part 1 – Introduction and general model
  - Part 2 – Security functional requirements
  - Part 3 – Security assurance requirements
- CEM – Evaluation Methodology
- Each country has a Scheme

Roles and responsibilities in CC

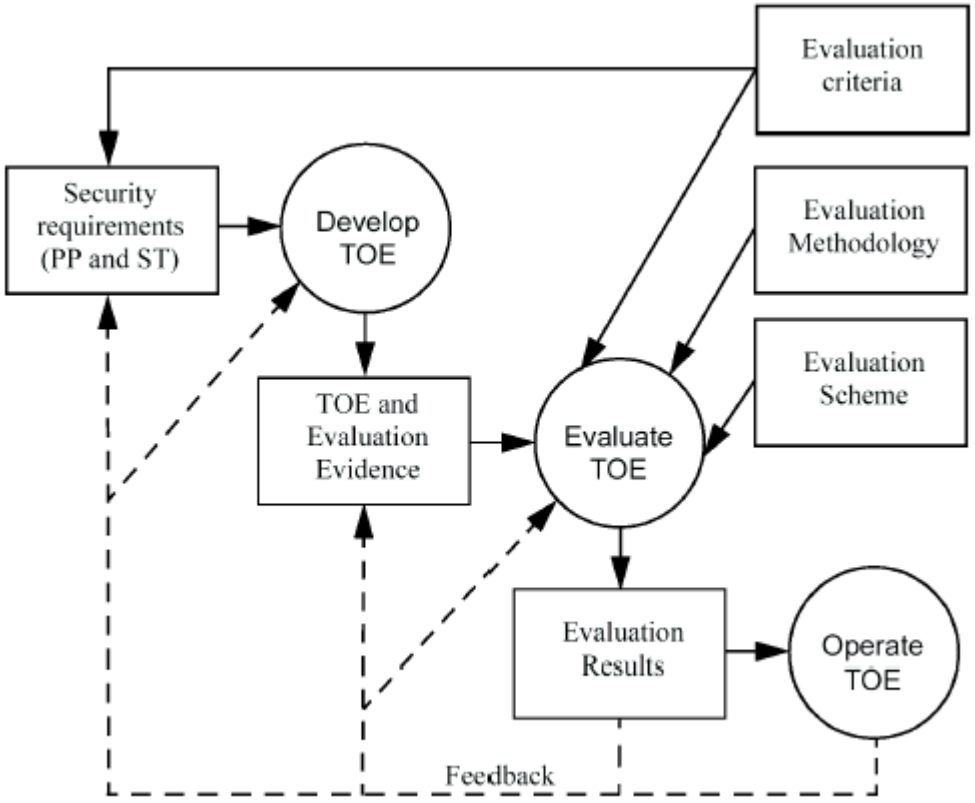


SECTRA

Terminology

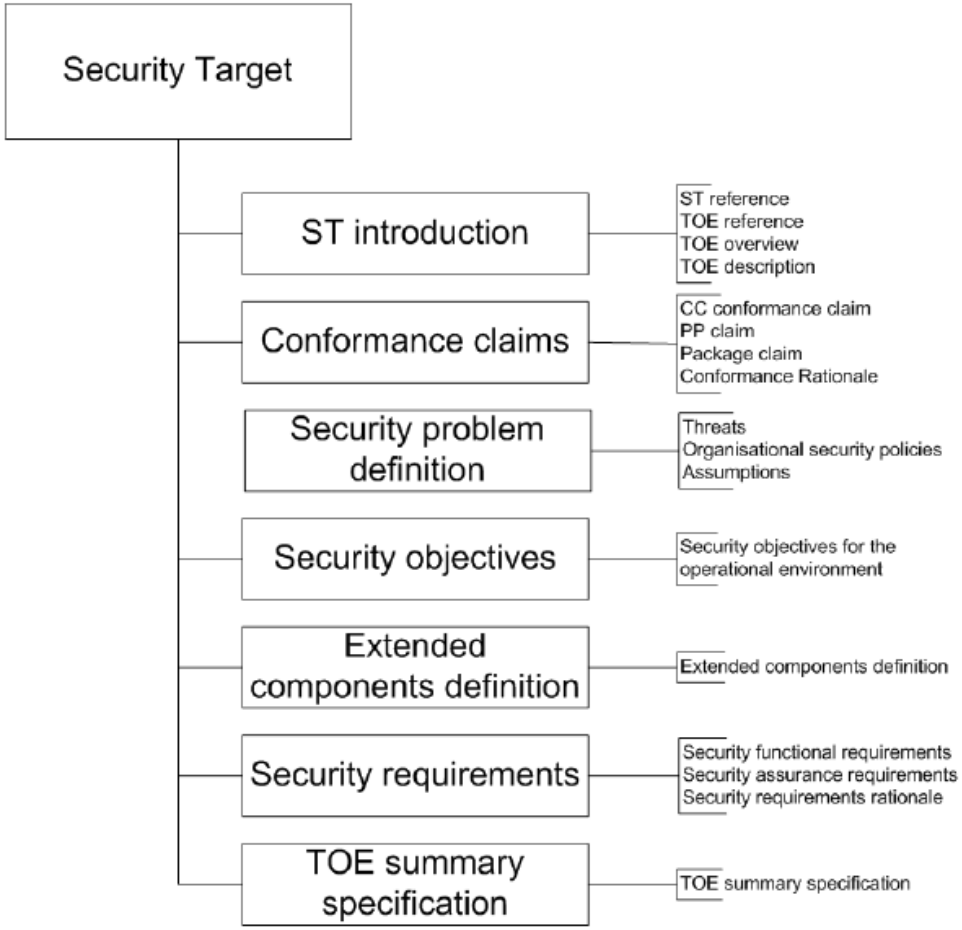
- Protection Profile (PP)
  - An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.
- Security Target (ST)
  - A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE
- Target of Evaluation (TOE)
  - The TOE is the entity, defined by the ST, that is evaluated
  - The TOE is the IT product or system, including the associated administrator and user guidance, that is the subject of an evaluation
- TOE Security Functionality (TSF)
  - The portions of the TOE that must be relied upon for the security enforcement

Evaluation Process



SECTRA

Security Target Structure



SECTRA

Structure of the Requirements

- A cookbook of predefined
  - Functional Requirements
  - Assurance Requirements
- Modular - classes, families, components, elements
- Hierarchy of components
- Dependencies between different components

Predefined Functionality Classes

- FAU – Security audit
- FCO – Communication
- FCS – Cryptographic support
- FDP – User data protection
- FIA – Identification and authentication
- FMT – Security management
- FPR – Privacy
- FPT – Protection of the TSF
- FRU – Resource utilization
- FTA – TOE access
- FTP – Trusted path/channels

Functional Requirement - Example

Security audit event storage (FAU_STG)

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to **[selection, choose one of: prevent, detect]** unauthorised modifications to the stored audit records in the audit trail.

Assurance Classes

Assurance Class	Assurance Family	Abbreviated Name
Class ADV: Development	Security Architecture	ADV_ARC
	Functional specification	ADV_FSP
	Implementation representation	ADV_IMP
	TSF internals	ADV_INT
	Security policy modelling	ADV_SPM
	TOE design	ADV_TDS
Class AGD: Guidance documents	Operational user guidance	AGD_OPE
	Preparative procedures	AGD_PRE
Class ALC: Life-cycle support	CM capabilities	ALC_CMC
	CM scope	ALC_CMS
	Delivery	ALC_DEL
	Development security	ALC_DVS
	Flaw remediation	ALC_FLR
	Life-cycle definition	ALC_LCD
	Tools and techniques	ALC_TAT
Class ASE: Security Target evaluation	Conformance claims	ASE_CCL
	Extended components definition	ASE_ECD
	ST introduction	ASE_INT
	Security objectives	ASE_OBJ
	Security requirements	ASE_REQ
	Security problem definition	ASE_SPD
	TOE summary specification	ASE_TSS
Class ATE: Tests	Coverage	ATE_COV
	Depth	ATE_DPT
	Functional tests	ATE_FUN
	Independent testing	ATE_IND
Class AVA: Vulnerability assessment	Vulnerability analysis	AVA_VAN

Evaluation Assurance Levels

Level EAL1
- The lowest level which should be considered for purposes of evaluation

Level EAL2
- Best that can be achieved without imposing some additional tasks on a developer

Level EAL3
- Allows a conscientious developer to benefit from positive security engineering design without alteration of existing reasonably sound development practices

Level EAL4
- The best that can be achieved without significant alteration of current good development practices.

Level EAL5
- The best achievable via pre-planned, good quality, careful security-aware development without unduly expensive practices.

Level EAL6
- A "high tech" level for (mainly military) use in environments with significant threats and moderately valued assets.

Level EAL7
- The greatest amount of evaluation assurance attainable whilst remaining in the real world for real products. EAL7 is at the limits of the current technology

Evaluation Packages and EAL Levels

Assurance class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Development	ADV_ARC		1	1	1	1	1	1
	ADV_FSP	1	2	3	4	5	5	6
	ADV_IMP				1	1	2	2
	ADV_INT					2	3	3
	ADV_SPM						1	1
	ADV_TDS		1	2	3	4	5	6
Guidance documents	AGD_OPE	1	1	1	1	1	1	1
	AGD_PRE	1	1	1	1	1	1	1
Life-cycle support	ALC_CMC	1	2	3	4	4	5	5
	ALC_CMS	1	2	3	4	5	5	5
	ALC_DEL		1	1	1	1	1	1
	ALC_DVS			1	1	1	2	2
	ALC_FLR							
	ALC_LCD			1	1	1	1	2
	ALC_TAT				1	2	3	3
Security Target evaluation	ASE_CCL	1	1	1	1	1	1	1
	ASE_ECD	1	1	1	1	1	1	1
	ASE_INT	1	1	1	1	1	1	1
	ASE_OBJ	1	2	2	2	2	2	2
	ASE_REQ	1	2	2	2	2	2	2
	ASE_SPD		1	1	1	1	1	1
	ASE_TSS	1	1	1	1	1	1	1
Tests	ATE_COV		1	2	2	2	3	3
	ATE_DPT			1	1	3	3	4
	ATE_FUN		1	1	1	1	2	2
	ATE_IND	1	2	2	2	2	2	3
Vulnerability assessment	AVA_VAN	1	2	2	3	4	5	5

SECTRA

Assurance Requirement - Example

ALC_CMC.1 Labelling of the TOE

ALC_CMC.1.1D (Developer action)

The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1C (Content and presentation)

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1E (Evaluator action)

The evaluator shall confirm that the Information provided meets all requirements for content and presentation of evidence.

Objective:

A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labelling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using.

CC Community

Certificate Authorizing
Australia and New Zealand
Canada
France
Germany
Italy
Japan
Malaysia
Netherlands
Norway
South Korea
Singapore
Spain
Sweden
Turkey
United Kingdom
United States

Certificate Consuming
Austria
Czech Republic
Denmark
Finland
Greece
Hungary
India
Israel
Pakistan
Singapore

SECTRA

Authorities in Sweden

CSEC	Certification body
ATSEC	Evaluation facility
Combitech	Evaluation facility
TSA	Swedish National Communication Security Agency, approval of cryptographic products
FMV	Swedish Defence Material Administration, sponsor of evaluation

SECTRA

Pros and Cons

+ Enforces a structural way of developing systems
+ Security is built into the system from the start
+ Becomes a natural part of the development process if done in the right way

- The documentation for the CC standard is extensive
- A costly process (time and money)
- Does not evaluate the technical solution

Recommendations

- Certify a well-known and relatively small product
- Start at a low assurance level, such as EAL2
- Go through a pre-evaluation if this is the first evaluation of the product
- Certify a product in development, changes to the product and its documentation are expected.

Recommendations

- Select a product that isn’t critical for time-to-market
- Select a product developed locally in one location
- Expect 4-6 months for EAL2 and about 1 year for EAL4
- The ST is a formal document and its quality is essential
- Do not write the ST yourself unless you have a strong CC background

Recommendations

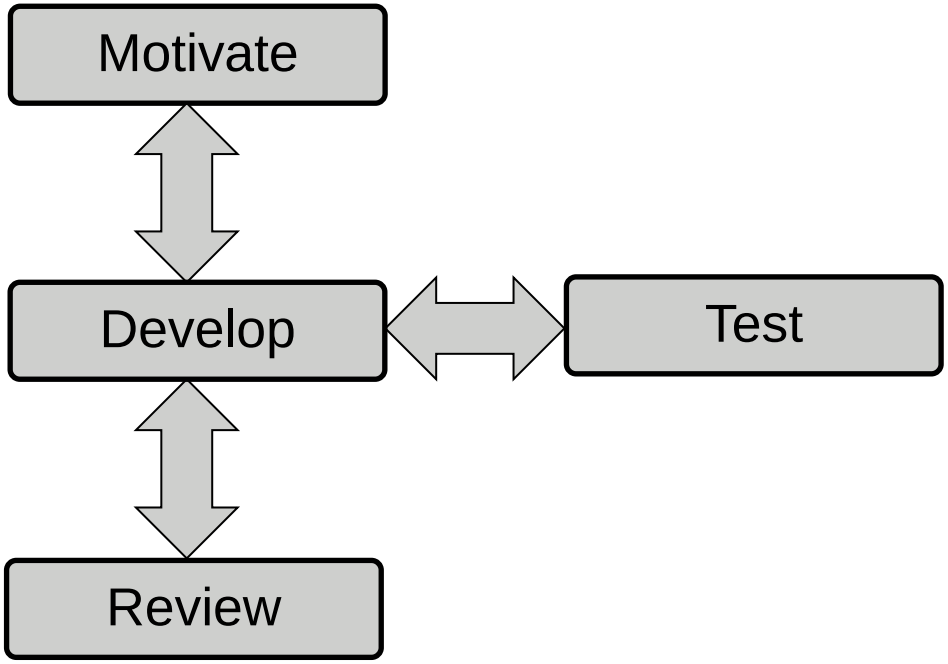
- Try to start the evaluation early in the development cycle
  - Makes it easier to include changes and bug fixes
- Document your processes and provide evidence that you follow them
- Use Configuration Management for everything

Break

Development Phases

- Preconditions
- Project definition
- System definition
- System design
- Implementation
- Verification and validation

Assurance



Preconditions

- Context of the system
- Primary assets
- Organisational policies
- Functional and security features
- Protection Profiles

- Threat analysis

SECTRA

Threat Analysis

- Assets
  - Attributes
  - Life-cycle
- Threat agents
  - Opportunity
  - Knowledge
  - Resources
  - Motivation
- Threats
  - Manipulation
  - Disclosure
  - Denial of service
- Countermeasures
- Policies
- Assumptions

Project Definition

- Time and activity planning
  - Delivery Plan (developer)
  - Evaluation Work Plan (evaluator)
- Define processes
  - Configuration management
  - Development security
  - Change management
  - Tools and techniques

- Evaluation of life-cycle management

System Definition

- Settle the requirements
  - Security Target
  - Functional requirements
  - Performance requirements

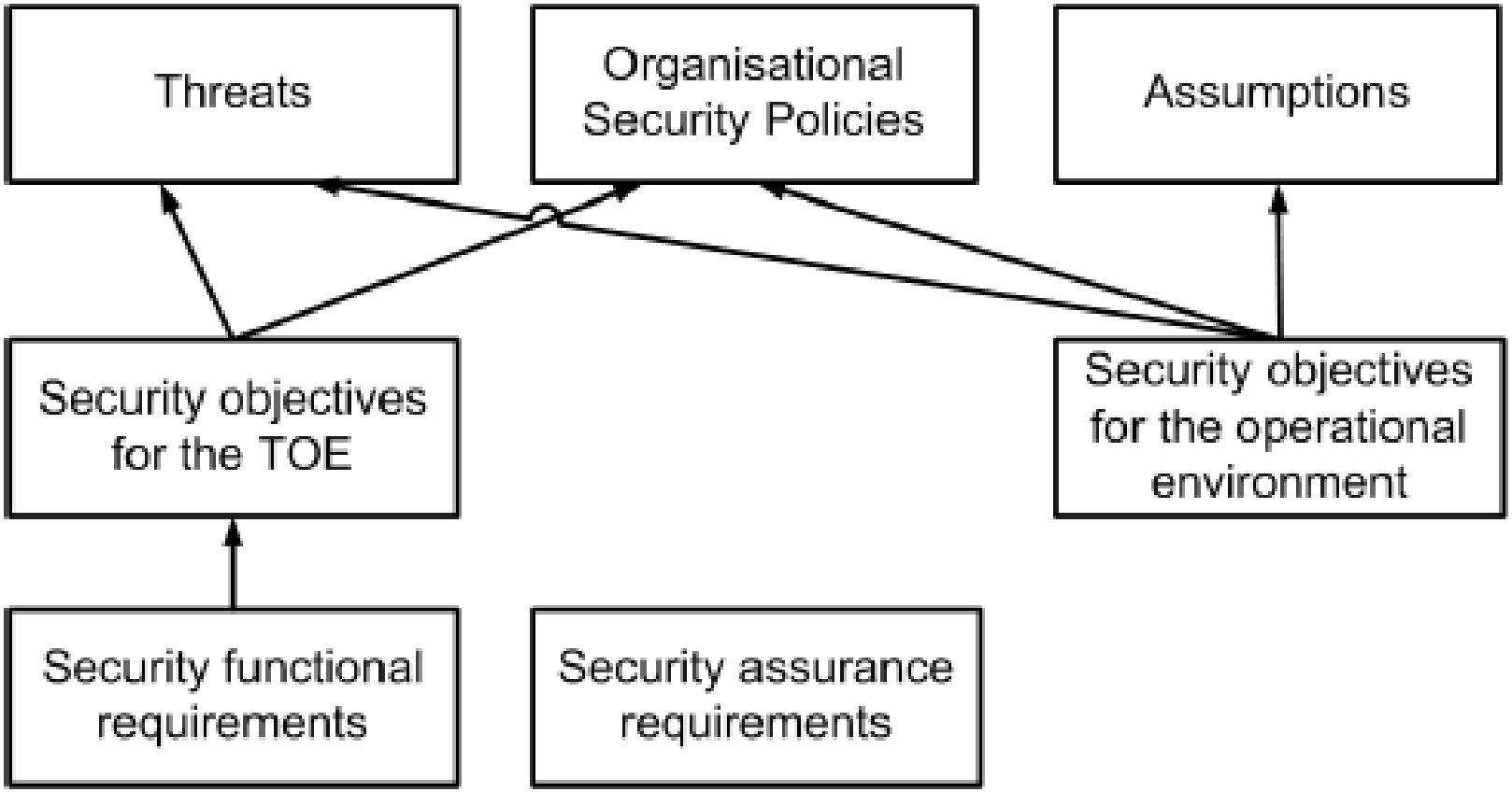
- Evaluation of ST

SECTRA

Security Target

- Security Problem Definition
  - Based on the threat analysis
- Security Objectives
- Security Functional Requirements
  - CC part 2
- Security Assurance Requirements
  - CC part 3
  - EAL-statement
- Justify the objectives and requirements

Security Objectives and Requirements



SECTRA

System Design

- Functional specification
  - Identifying security functions
- External interfaces
  - Identifying interfaces to security functions
- Design
  - Decomposition
  - Dependencies
  - Dynamic behavior
  - Map security functions

- Evaluation of the design

Implementation phase

- Detailed design
- Map security functions
- Tag the security functions in the implementation

- Evaluate the detailed design and implementation

Verification and Validation

- Verify the design
- Validate the fulfillment of requirements

- Evaluate the tests
- Independent test by the evaluator
- Vulnerability analysis and penetration tests
- Evaluate the guidance documentation

Gaining trust

- Reviews
- Tests
- Security Architecture
- Correspondence analysis
- Physical and logical protection analysis
- Security verification

Reviews

- Document review
- Implementation review
- Pair programming

- Evidence of reviews

Test

- Unit test
- Integration test
- System test
- Penetration test
- Fuzz test

- Test coverage
- Code coverage
- Evidence of tests
- Depth of tests

Security Architecture

- Explains how the system is designed and implemented concerning the following three aspects:
  - Self-protection: How is the integrity of the security functionality in the system preserved?
  - Protection concerning bypass: How is security functionality in the system protected from being bypassed?
  - Domain separation: How is the system divided into different security domains?
- The level of detail is given by the assurance requirements

Correspondence Analysis

- Justification of the fulfillment of security functional requirements
- Justification of the interfaces to the security functions
- Justification of security function decomposition

Physical and Logical Protection Analysis

- Analysis of the implementation of the security functions
  - Does the physical implementation counter the threats?
  - Does the interfaces counter logical attacks?
- Attacker perspective

SECTRA

Security Verification

- Verify the cryptographic mechanisms
- Verify the security functionality

www.commoncriteriaportal.org

SECTRA