DEG 203 350 v0.0.4 (2014-04-11)

Methods for Testing and Specification (MTS);

Security Assurance Lifecycle

<

**ETSI GUIDE**

Reference

DEG/MTS-203250

Keywords

Security; Testing; Assurance

***ETSI***

650 Route des Lucioles

F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C

Association à but non lucratif enregistrée à la

Sous-Préfecture de Grasse (06) N° 7803/88

***Important notice***

Individual copies of the present document can be downloaded from:  
<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:  
<http://portal.etsi.org/chaircor/ETSI_support.asp>

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute yyyy.

All rights reserved.

**DECT**TM, **PLUGTESTS**TM, **UMTS**TM and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.  
**3GPP**TM and **LTE**™ are Trade Marks of ETSI registered for the benefit of its Members and  
of the 3GPP Organizational Partners.  
**GSM**® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Reproduction is only permitted for the purpose of standardization work undertaken within ETSI.  
The copyright and the foregoing restriction extend to reproduction in all media.

Contents

Intellectual Property Rights 6

Foreword 6

1 Scope 7

2 References 8

2.1 Normative references 8

2.2 Informative references 8

3 Definitions, symbols and abbreviations 10

3.1 Definitions 10

3.2 Symbols 10

3.3 Abbreviations 10

4 Security in the standardisation lifecycle 11

4.1 Security and context 11

4.2 Defintion of Lifecycle 11

4.3 Security Architectures 17

4.4 Lifecycle stage component activities 18

4.4.1 Stakeholder Requirement Definition 19

4.4.2 Requirements Analysis 19

4.4.3 Architectural Design 19

4.4.4 Implementation 20

4.4.5 Integration 20

4.4.6 Verification 20

4.4.7 Transition 20

4.4.8 Validation 21

4.4.9 Operation 21

4.4.10 Maintenance 21

4.4.11 Disposal 21

4.5 Lifecycle-specific security characterictics 22

5 Security design activities 23

5.1 Approach 23

5.2 Design dependencies 23

5.3 Security design and implementation processes 23

5.3.1 Adversity Analysis (AA) 23

5.3.2 Risk Analysis (RA) 23

5.3.3 Vulnerability Analysis (VA) 23

5.3.4 Architectural Reference Model (ARM) 23

5.3.5 Architectural Reference Case (ARC) 24

5.3.6 Architectural Specification Case (ASC) 24

5.3.7 Design and/or Effect Classes 24

5.3.8 System Design 24

5.3.9 Traceable Implementation 24

5.3.10 Integration and Configuration 24

5.4 Assurance and Testing processes 24

5.4.1 Testing 24

5.4.2 Assurance 24

5.5 Post-implementaion processes 25

5.6.1 Commissioning 25

5.6.2 Operation 25

5.6.4 Disposal 25

Annex A (informative): Security contexts and principles 26

Annex B (informative): Security Architectural Components 27

Annex C (informative): Bibliography 31

History 32

# Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

# Foreword

This final draft ETSI Guide (EG) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS), and is now submitted for the ETSI standards Membership Approval Procedure.

# 1 Scope

The present document provides a guide to the application of security capabilities in systems in such a way to maximise both security assurance and the verification and validation of the capabilities offered by the systems's security measures.

The present document extends the guidance given in EG 202 387 [i.1], TS 102 165‑1 [i.2], TR 187 023 [i.3] and TS 101 583 [i.4] in a manner that brings the core understanding of security provisions for the purpose of testing in the meaning of verification and validation of assurance. The present document considers the standardisation domain and gives examples of the application of the guidance to the wider development and delployment community.

# 2 References

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references,only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

## 2.1 Normative references

The following referenced documents are necessary for the application of the present document.

Not applicable.

## 2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI EG 202 387: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Security Design Guide; Method for application of Common Criteria to ETSI deliverables".

[i.2] ETSI TS 102 165-1 (V4.2.1): "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis".

[i.3] ETSI TR 197 023: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Security Assurance Profile for Secured Telecommunications Operations"

[i.4] ETSI TS 101 583: “Security Testing Terminology and Concepts”

[i.5] ISO/IEC 12207-2008: “Systems and software engineering -- Software life cycle processes”

[i.6] ISO/IEC 15288-2008: “Systems and software engineering -- System life cycle processes”

[i.7] BSI PAS nnn (201y): “Trustworthy Software Framework”

[i.8] ISO/IEC 15026-2-2011: “Systems and software engineering -- Systems and software assurance -- Part 2: Assurance case”

[i.9] John A. Zachman “A Framework for Information Systems Architecture“, IBM Systems Journal 1987 (G321-5298), Vol 26, No 3

[i.10] US DOD Technical Architecture Framework for Information Management (TAFIM), 1991

NOTE: Later available as Command, Control, Communications, Computers, and Intelligence, Surveillance, and Reconnaissance (C4ISR), Jun-1996, and Department of Defense Architecture Framework (DoDAF), Aug-2003

[i.11] IFAC/IFIP Generalised Enterprise Reference Architecture and Methodology (GERAM), 1994

[i.12] The Open Group Architecture Framework (TOGAF), v1.0, 1995

[i.13] Reference Model of Open Distributed Processing (RM-ODP) (ITU-T Rec. X.901-X.904 & ISO/IEC 10746), 1998

[i.14] Enterprise Architecture Planning (EAP), 1998

[i.15] US ANSI/IEEE 1471 Recommended practice for architectural description of software-intensive systems, September 2000

NOTE: Now available as ISO/IEC 42010:2007

[i.16] UK Ministry of Defence Architecture Framework (MoDAF), Aug-2005

[i.17] NATO Architecture Framework (NAF),

NOTE: This is a fusion of DoDAF and MoDAF extebded for the NATO context.

# 3 Definitions, symbols and abbreviations

## 3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply:

**<defined term>:** <definition>

**example 1:** text used to clarify abstract rules by applying them literally

NOTE: This may contain additional information.

## 3.2 Symbols

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

<symbol> <Explanation>

<2nd symbol> <2nd Explanation>

<3rd symbol> <3rd Explanation>

## 3.3 Abbreviations

For the purposes of the present document, the [following] abbreviations [given in ... and the following] apply:

<ACRONYM1> <Explanation>

<ACRONYM2> <Explanation>

<ACRONYM3> <Explanation>

# 4 Security in the system lifecycle

## 4.1 Security and context

Security is most often defined as being the ability of a system to give assurance of the following 3 key attributes of a product or service:

* Confidentiality
* ensuring that information is accessible only to those authorized to have access
* Integrity
* safeguarding the accuracy and completeness of information and processing methods
* Availability
* property of beingaccessible and usable on demand by an authorized entity ISO/IEC 13335-1 [17].

Underpinning these attributes are the contained attributes authenticity and authority, and underpinning these is identity. It should be recognised that a single definition of security is largely infeasible without also defining its context and as such to say that a system of product is secure is wrong without defining in some detail the context in which that assurance is given. This differentiates the tools for verification and validation in a security context from the same tools used in a conventional state-machine based model as failure in the security of a system may not be directly related to a specific element, e.g. the authentication protocols used in a system may provably work but the system may be broken if the authentication protocols can be bypassed, thus proof of the correctness of a security measure may be insufficient to give assurance of the security of the system.

## 4.2 Definition of Lifecycle

The system design lifecycle described in TS 102 165-1 [2] and replicated here identifies, for a general system design, the processes and outputs of the design lifecycle.



Figure 1: Structure of security analysis and development in standards documents

The purpose of the TVRA is to determine how open to attack the system, or components of the system are. The resulting security architecture and security mechanisms are intended to maintain the system in a state of acceptable risk by removing the set of vulnerabilities that when attacked will violate the level of acceptable risk. The alternative view of the nature of TVRA is given in figure showing that any change either internal (say by application of countermeasures) or external to the system requires that the TVRA process is redone. The role of verification and validation in system security is primarily to give assurance that the designed for risk profile is maintained over time.



Figure 2: Cyclical nature of TVRA

As noted in TS 102 165-1 [] one of the keys to a successful TVRA, and also of a successful system design, is the ability to show the relationship of objectives and requirements to the system design. Figure shows the dependencies between system objectives, system requirements and system design highlighting the interplay of security objectives and requirements.



Figure 3: Relationship between system design, objectives and requirements

For most systems the development of system requirements goes far beyond just security it is essential to ensure that the system design is itself robust and therefore has fully documented requirements across all its aspects. The system design and its analysis requires that the assets of the system, the relationship of the system to its environment are clearly identified (this is essential in showing the contextual nature of security). A pictorial view of the asset-threat-weakness-vulnerability-countermeasure relationship to system design is given in figure from TS 102 165-1 [2].



Figure 4: Generic security TVRA model

One of the purposes of security design is to minimize the probability of any instance of the class "unwanted incident" being instantiated. It should be noted that whilst some countermeasures may themselves become system assets, and as such have their own vulnerabilities, many instances of countermeasures will be considered as policies, system guidelines and, if captured early enough, system redesign. The threats to the assets of the system can be classified as one of 4 types:

* Interception.
* Manipulation.
* Denial of service.
* Repudiation of sending / Repudiation of receiving.

Similarly security objectives can be classified as one of 5 types (commonly referred to as "CIAAA" types):

* Confidentiality.
* Integrity.
* Availability
* Authenticity (requires Identity)
* Accountability (requires Authority).

The primary purpose of an ETSI TVRA is to support and rationalize security standardization, and to support and rationalize system design decisions, where the overall objective of the standard is to minimize risk of exploitation and attack of a compliant system when deployed. In order to consider this fully the TVRA method described in the present document addresses the impact of an attack on the system whereas ISO-15408 [13] primarily addresses the resistance to attack of the system. In this view the TVRA method compliments ISO-15408 [13]. A particular objective of the TVRA method is to prepare the justifications for security decisions and that may as a result be referenced in a PP for the security feature.

The conventional standards development lifecycle for protocols defined by ITU-T in I.130 and I.210 specify a 3-stage process:

* Stage 1: Specify requirements from the user's perspective;
* Stage 2: Develop a logical model to meet those requirements;
* Stage 3: Develop a detailed specification of the protocol.

Whilst I.130 requires that for each stage there is a formal document as input for the next stage the form of the "document" can be varied to, for example, different views of a single UML model. The staged process is also extended by adding validation and verification between stages (e.g. to verify that requirements are achievable) and to achieve tracability (e.g. to show how the protocol meets the user requirements)

Before considering the detailed specification of a protocol, there are benefits to be gained by specifying a model based on logical blocks so that the flow of information necessary for meeting the specified requirements can be defined without concern for the detailed format that such information should take. The identification of possible normative interfaces between blocks is also simpler without the constraints imposed by a specific physical architecture.

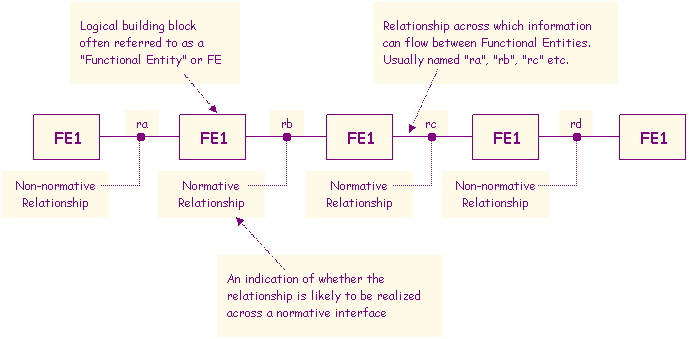


Figure 5: Generalised relationshop model

The entity relationship diagram as shown above, together with information flow diagrams (sequence diagrams) are the most important elements of the logical model. The behaviour of each FE may be specified with UML or SDL state charts. However, stage two behaviour descriptions should remain significantly simpler then stage 3 behaviour descriptions.

There are many definitions of the Lifecycle for a product or service, from the very simple (Specification – Realisation – Use) to the complex. Probably the most widely accepted are those produced by ISO/IEC JTC1 SC7 as defined in ISO/IEC 12207 [i.5] and ISO/IEC 15288 [i.6].

A mapping of security component activities to the ISO/IEC 15288 Lifecycle is summarised in .



Figure 6: Security design lifecycle activity diagram

The 11 stage model given in Figure 6 extends the activity beyond standardisation and through the product or service lifecycle of which the specification of the product in standards is only a small part of the lifecycle. However the impact of standardisaiton on each of the 11 phases is discussed in more detail below.

1. Stakeholder Requirement Definition
2. Requirements Analysis
3. Architectural Design
4. Implementation
5. Integration
6. Verification
7. Transition
8. Validation
9. Operation
10. Maintenance

Maintenance can be considered to be a “mini-lifecycle”, potentially reflecting all preceding stages, and to be carried out successfully access to artefacts from the original development lifecycle will be a fundamental dependency.

1. Disposal

## 4.3 Security Architectures

In terms of ICT, high level architecture is generally accepted to first introduced as the Zachman Framework [i.9], and has gone through many iterations since including those identified in the US DOD Technical Architecture Framework for Information Management (TAFIM) [i.10], IFAC/IFIP Generalised Enterprise Reference Architecture and Methodology (GERAM) [i.11], The Open Group Architecture Framework [i.12], Reference Model of Open Distributed Processing [i.13], Enterprise Architecture Planning [i.14], US ANSI/IEEE 1471 Recommended practice for architectural description of software-intensive systems [i.15], UK Ministry of Defence Architecture Framework [i.16], and NATO Architecture Framework [i.17].

Architectural Frameworks typically decompose an architecture into a number of Levels, with a common set being:

* Level 1 - Conceptual
* Level 2 - Contextual
* Level 3 - Logical
* Level 4 - Physical
* Level 5 - Detailed

Interpreting these levels for Information Security to cover both Requirements and Implementation, leads to a Security Architecture Framework (SAF) as shown in Figure 7.

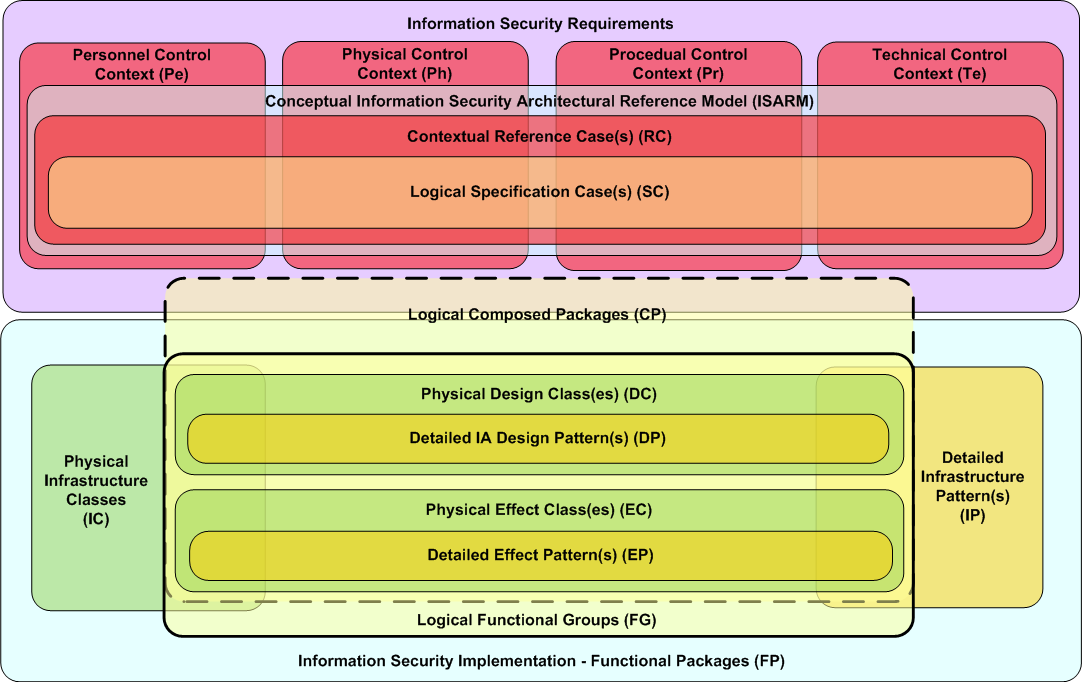


Figure 7: Security Architecture Framework

In this model Generic functional specifications are defined as a CLASS and detailed implementations defined as PATTERNS. A collection of Patterns is referred to as a Composed Package.

* An import distinction in this model is between DESIGN and EFFECT:
* DESIGNS define requirements for components provide architectural Security Enforcing Functions (SEF), with Designs Classes / Patterns for Infrastructure (IC/ IP) being a special case which define SEFs for use as common Enterprise Architecture (EA) infrastructure components
* EFFECTS define IA requirements for components where security is a Non-Functional Requirement (NFR)

## 4.4 Lifecycle stage component activities

From consideration of both the lifecycle mapping of security activities and Security Architecture Framework (SAF), a relevance mapping has been produced, covering both Architecture and Assurance activities, as provided at Figure 8, which also shows the minor variances between products and services.

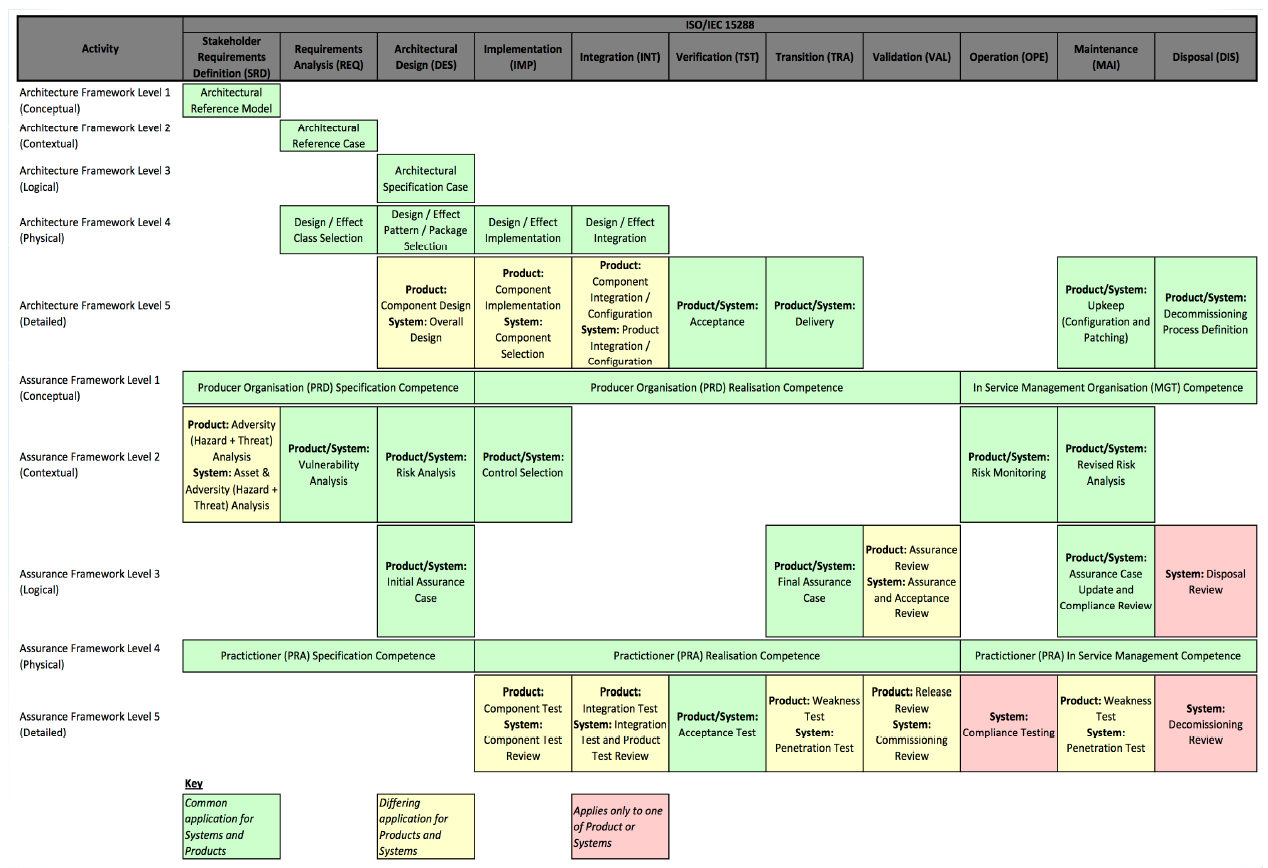
****

Figure 8: Security activity relevance mapping

### 4.4.1 Stakeholder Requirement Definition

* Product or System Architecture
  + Development or selection of Architectural Reference Model (ARM)
* Product Assurance
  + Performance of Adversity (Hazard + Threat) Analysis
* System Assurance
  + Asset & Adversity (Hazard + Threat) Analysis

### 4.4.2 Requirements Analysis

* Product or System Architecture
  + Development of Architectural Reference Case (ARC)
  + Selection of appropriate Design and/or Effect Class(es)
* Product or System Assurance
  + Performance of Vulnerability Analysis

### 4.4.3 Architectural Design

* Product or System Architecture
  + Development of Architectural Specification Case (ASC)
  + Selection of appropriate Design and/or Effect Pattern(s) or Package(s)
* Product Architecture
  + Development of Component Design
* System Architecture
  + Development of System Design
* Product or System Assurance
  + Performance of Risk Analysis
  + Development of Intitial Assurance Case

### 4.4.4 Implementation

* Product or System Architecture
  + Implementation of selected Security Design(s) and/or Effect(s)
  + Selection of required Components
* Product Architecture
  + Implementation of required Components
* Product Assurance
  + Developer test of implemented Components
* System Assurance
  + Developer review of testing of implemented Components

### 4.4.5 Integration

* Product or System Architecture
  + Integration of implemented Security Design(s) and/or Effect(s)
  + Integration and configuration of implemented Components
* System Architecture
  + Integration and configuration of implemented Products
* Product Assurance
  + Integration test of implemented Components
* System Assurance
  + Integration review of testing of implemented Components and Products

### 4.4.6 Verification

* Product or System Assurance
  + Acceptance Test

### 4.4.7 Transition

* Product or System Architecture
  + Delivery
* Product or System Assurance
  + Finalisation of Assurance Case
* Product Assurance
  + Performance of Vulnerability Test
* System Assurance
  + Performance of Susceptability (“Pentration”) Test

### 4.4.8 Validation

* Product Assurance
  + Performance of Assurance Review
  + Performance of Release Review
* System Assurance
  + Performance of Assurance and Acceptance Review
  + Performance of Commissioning Review

### 4.4.9 Operation

* Product or System Assurance
  + Risk Monitoring
* System Assurance
  + Performance of Compliance Tests

### 4.4.10 Maintenance

Maintenance can be considered to be a “mini-lifecycle”, potentially reflecting all preceding stages, and to be carried out successfully access to artefacts from the original development lifecycle will be a fundamental dependency.

In particular, considation is needed of:

* Product or System Architecture
  + Regular upkeep (Configuration and Patching)
* Product or System Assurance
  + Revision of Risk Analyses
  + Update of Assurance Case
  + Performance of Compliance Reviews
* Product Assurance
  + Performance of Assurance and Acceptance Review
  + Performance of Vulnerability Test
* System Assurance
  + Performance of Susceptability (“Pentration”) Test

### 4.4.11 Disposal

* Product or System Assurance
  + Definition of Decommissioning Process
  + Update of Assurance Case
  + Performance of Compliance Reviews
* System Assurance
  + Decommissioning Review
  + Disposal Review

## 4.5 Lifecycle-specific security characterictics

In order to achieve security through an extended lifecycle, a fundamental requirement is for Traceability, with a need for continuity of access to artefacts from earlier lifecycle stages.

In addition, due consideration is needed of Composability, with an understanding of the Risks and Dependencies, Assumptions and Assertions introduced as atomic components are composed into compound packages, packages are composed in compound products, products are composed into compound systems, and systems are composed into compound systems-of-systems.

# 5 Security design activities

## 5.1 Approach

In order to achieve Security, it needs to be considered in 4 contexts:

* Governance – the processes and people needed for properly administering secure delivery of a product or service
* Risk – the factors which could have a deleterious impact on secure delivery of a product or service
* Controls – the principles and technqiues used to reduce the risk to secure delivery of a product or service, which can be further subdivided as:
* Personnel Controls
* Physical Controls
* Procedural Controls
* Technical Controls
* Compliance – the processes and people needed for ensuring secure delivery of a product or service is proceeding as intended

A summary of Principles for realisation of secure products and services, as drived from UK’s Trustworthy Software Framework (TSF) [i.7] is provided at Annex A.

## 5.2 Design dependencies

In line with the lifecycle definition, it is assumed that before commencing a fresh design for a product or service, Stakeholder Requirements Definition will have been completed.

Design will also be a phase during the Maintenance “mini-lifecycle”, and to be carried out successfully access to artefacts from the original development lifecycle will be a fundamental dependency.

## 5.3 Security design and implementation processes

The lifecycle mapping of security activities and Security Architecture Framework (SAF) identifies the following activitities for system design and implementation in support of Test and Assurance.

### 5.3.1 Asset and Adversity Analysis (AA)

In order to understand the requirements for security activities, the underlying business drivers first need to be analaysed.

This requires the enumeration and collation of:

* Assets – establishing both the Utility to the Service Provider of the target infrastructure(s) to be secured, and the Utility of Information stored, processed or forwarded on these infrastructures to the customer(s)
* Adversity – establishing both the set of Threats (from malicious intent, such as Hackers) and Hazards (from undirected factors such as Electromagentic Interference of both man-made sources and terrestrial / extra- terrestrial natural sources)

### 5.3.2 Risk Analysis (RA)

Having established the set of Asset Utilities and Adversities faced, the probability of occurance of Adversities needs to be analaysed, and the potential impact established based on the assessed Asset Utilities, yielding a set of Risks.

These Risks should then be grouped, as far as possible, into potential common Compromise categories that need to be mitigated.

### 5.3.3 Vulnerability Analysis (VA)

Having established the Compromises that need to be avoided, the target solution elements should be enumerated, and the known instantiation of any generic Weakness (for instance unpatched Buffer Overflows) listed.

### 5.3.4 Architectural Reference Model (ARM)

The Architectural Reference Model (ARM) is a high level generic specification of functional components of an infrastructure(s) that are reused within a solution delivery organisation, defining the security characteristics, assumptions and constraints of functional components.

This model should be maintained in an interative manner, with lessons identified during each solution implementation used to update the list of security characteristics, assumptions and constraints.

### 5.3.5 Architectural Reference Case (ARC)

The system design for the specific solution should be mapped onto the relevant subset of the Architectural Reference Model (ARM), to produce a Architectural Reference Case (ARC) which summarises only those functional components that are applicable.

### 5.3.6 Architectural Specification Case (ASC)

The Architectural Reference Case (ARC) should be used to produce a master collation of security characteristics, assumptions and constraints relevant to an implementation, referred to as the Architectural Specification Case (ASC).

### 5.3.7 Design and/or Effect Classes

The Architectural Specification Case (ASC) should be used to select the relevant Design and/or Effect classes needed to achieve the security characteristics for a solution. A list of Design and/or Effect classes is provided at Annex B.

### 5.3.8 System Design

The security aspects of the system design process should be, as far as possible, driven by the list of the relevant Design and/or Effect classes needed to achieve the security characteristics for a solution, with only special features needing to be specifically designed.

Such reuse of established principles and techniques can both significantly reduce the effort required for design, and improve the ability to assure the solution by the ability to reuse previous evidence.

The implementation should factor in mitigation of all Compromises identified by Risk Analysis.

### 5.3.9 Traceable Implementation

The implementation of the design should be clearly traceable back to the set of Design and/or Effect classes needed to achieve the security characteristics, and additional special features, in order to provide traceability of the delivery of the security requirement.

The implementation should factor in mitigation of all issues identified by Vulnerability Analysis.

### 5.3.10 Integration and Configuration

The integration and configuration of the solution should be derived from the set of security characteristics, assumptions and constraints relevant to an implementation, with the principle of Least Privelige being applied.

The implementation should factor in mitigation of all issues identified by Vulnerability Analysis that were not addressed during Implementation.

## 5.4 Assurance and Testing processes

The lifecycle mapping of security activities and Security Architecture Framework (SAF) identifies the following activitities for Assurance and Testing.

### 5.4.1 Testing

<< Reference out to TS 101 583 >>

<< - Component >>

<< - Integration >>

<< - Acceptance >>

<< - Vulnerability Test >>

<< - Susceptability (“Penetration”) Test >>

### 5.4.2 Assurance

<< This is where we add guidance on how to perform Assurance >>

<< Reference to Assurance Case [i.8] >>

<< Include Assurance and Acceptance Review >>

<< Include Release Review >>

## 5.5 Post-implementaion processes

The lifecycle mapping of security activities and Security Architecture Framework (SAF) identifies the following activitities as impinging on security, after intial Assurance has been achieved.

### 5.6.1 Commissioning

<< This is where we add guidance on how to perform >>

### 5.6.2 Operation

<< This is where we add guidance on how to perform Compliance Reviews (incl. Vulnerability Test and Susceptability (“Penetration”) Test), Regular upkeep (Configuration and Patching), >>

### 5.6.4 Disposal

<< This is where we add guidance on how to design a Decommissioning Process, including Reviews >>

Annex A (informative):  
Security contexts and principles

|  |  |
| --- | --- |
| **Context** | **Principles** |
| Governance | Understand General Environment |
| Understand Threat Environment |
| Implement formal management regime |
| Risk | Understand General Risks |
| Understand Security Risks |
| Controls - Personnel | Maintain Practitioner Competence |
| Maintain Organisational Competence |
| Controls - Physical | Protect Physical Environment |
| Provide Artefact Protection |
| Controls - Procedural | Perform Project Management |
| Perform Supplier Management |
| Understand Requirements |
| Maintain Configuration Management |
| Confirmation of Assurance |
| Perform Trusted Software Asset Management |
| Maintain Fault Management |
| Controls - Technical | Follow Architecture-driven Implementation |
| Make appropriate tool choices |
| Follow Structured Design |
| Follow Structured Implementation |
| Seek Trustworthy Realisation |
| Minimise risk exposure |
| Practice Hygienic Coding |
| Use Methodological Implementation |
| Perform Internal Pre-release Review |
| Perform Internal Verification |
| Enable Dependable Deployment |
| Compliance | Perform Independent Verification |
| Maintain Ongoing Review |

Annex B (informative):  
Security Architectural Components

**Functional Groups and Classes**

**1. Functional Groups (12)**

Most abstract – Logical Level – decomposition of the Implementation Segment of the Security Architectural Framework is into IA Functional Groups (IAFG):

a. IA Technical Functions (8)

* Connectivity Protection (Te - Conn)
* Cryptography (Te - Cryp)
* Entity Authorisation (Te - EntA)
* Media & Device Protection & Disposals (Te - MDPD)
* Monitoring and Surveillance (Te - MonS)
* Information Integrity Preservation & Protection (Te - IIPP)
* Media and Information Authentication (Te MIAu)
* Intrinsic ICT Functions (Te - IICT)

b. IA Non-Technical( “P3”) Functions (3)

* Personnel Security Functions (Pers)
* Physical Security Functions (Phys)
* Procedural Functions (Proc)

c. Data Management Functions (1)

* Security Labels

**2. Classes (69)**

a. IA Connectivity Protection (Te – Conn (6))

* Perimeter Control Design Class
* Virtual Connection Design Class
* Wireless Protection Design Class
* Port Protection Effects Class
* WAN/MAN Protection Effects Class
* LAN/PAN Protection Effects Class

b. IA Cryptography (Te – Cryp (7))

* Credential Generation
* Credential Management
* Credential Storage
* Network Encryption
* File Encryption
* Internal Media Encryption
* Removable Media Encryption

c. IA Entity Authorisation (Te – EntA (8))

* User Authentication
* Multifactor Authentication
* Privilege Management
* Biometrics
* ID Management
* Federation
* Session Authentication
* User Registration

d. IA Media & Device Protection & Disposals (Te – MDPD (4))

* Media Erasure
* Object Erasure
* Mobile device remote disable/erase
* Hard Copy Protection

e. IA Monitoring and Surveillance (Te – MonS (8))

* Intrusion Detection
* Intrusion Protection
* Protective Monitoring
* Content Scanning (Malware / SpyWare)
* Forensic Capture
* Network Devices
* Network Management
* Mobile device tracking

f. IA Information Integrity Preservation & Protection (Te – IIPP (11))

* Device / Executable Control
* Technical Vulnerability Management
* Information Management
* Code Management
* Pre Boot controls
* System Virtualisation
* Time synchronisation
* Software licensing
* Offline Backup
* Remote Backup
* BCP / DR Facilities

g. IA Media and Information Authentication (Te MIAu (2[[1]](#footnote-1)))

* Media Authentication
* Realtime Policy Advice / Enforcement

h. Intrinsic ICT Functions (Te – IICT (12))

* Generic Effects Class
* Server Effects Class
* Workstation Effects Class
* Web Browser Effects Class
* Web Server Effects Class
* Email Server Effects Class
* DNS Server Effects Class
* PED Effects Class
* Service Bus Effect Class
* Audio Device Effect Class
* Video Device Effect Class
* Database Effect Class

i. Personnel Security Functions (Pers (3))

* Awareness
* Training
* Education

j. Physical Security Functions (Phys (3))

* Data Centre Security
* Static UAD Security
* Mobile UAD Security

k. Procedural Security Functions (Proc (5))

* Audit Analysis
* Forensic Analysis
* Assurance Testing
* Supply Chain Trust
* BCP/DR testing

l. Data Management Functions (1)

* Security Labels

Annex C (informative):  
Bibliography

[i.4] ISO/IEC 13335: "Information technology - Guidelines for the management of IT security".

[i.5] AS/NZS 4360: "Risk Management".

[i.6] Directive 2002/21/EC of the European Parliament and of the council of 7 March 2002 on a common regulatory framework for electronic communications networks and services (Framework Directive).

[i.7] Directive 2002/58/EC of the European Parliament and of the council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications).

[i.8] ISO/IEC 15408-2: "Information technology - Security techniques - Evaluation criteria for IT security - Part 2: Security functional requirements".

[i.9] ISO/IEC 15408: "Information technology - Security techniques - Evaluation criteria for IT security".

NOTE: When referring to all parts of ISO/IEC 15408 the reference above is used.

[i.10] ETSI TR 187 011: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Security; Application of ISO-15408-2 requirements to ETSI standards - guide, method and application with examples".

[i.11] Directive 95/46/EC Of The European Parliament And Of The Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

[i.12] UK Home Office; R.V.Clark, "Hot Products: understanding, anticipating and reducing demand for stolen goods", ISBN 1-84082-278-3.

[i.13] ISO/IEC 7498-2: " Information processing systems -- Open Systems Interconnection -- Basic Reference Model -- Part 2: Security Architecture".

[i.14] ETSI TS 102 165-2: " Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 2: Protocol Framework Definition; Security Counter Measures"

# History

|  |  |  |
| --- | --- | --- |
| **Document history** | | |
| V0.0.1 | November 2013 | Initial Draft for MTS Security SIG: Ian Bryant <mailto:ian.bryant@uk-tsi.org> |
| V0.0.2 | December 2013 | Updated Draft after MTS Security SIG comments: Ian Bryant <mailto:ian.bryant@uk-tsi.org> |
| V0.0.3 | January 2014 | Updated Draft after with Design detail (section 5.3): Ian Bryant <mailto:ian.bryant@uk-tsi.org>  Renumbered from DEG 202 792 to DEG 203 250 |
| V0.0.4 | April 2014 | Marked-up Draft for discussion by MTS Security SIG:  Ian Bryant <mailto:ian.bryant@uk-tsi.org> and Jürgen Großmann <mailto:juergen.grossmann@fokus.fraunhofer.de> |

1. Logically also includes 3rd Class, Information Labelling, but this is now broken out as own Functional Group – Security Labelling [↑](#footnote-ref-1)