|  |
| --- |
|  |
| **Title\*:** | Modelling and design for security |
|  |  |
| from **Source**\*: | Cadzow Communications |
| Contact: | Scott Cadzow  |
|  |  |
| input for **Committee**\***:** | MTS |
|  |  |
| Contribution **For\*:** | Decision |  |  |
|  | Discussion | **X** |  |
|  | Information |  |  |
|  |  |
| Submission date**\***: | 2011-12-10 |
|  |  |
| Meeting & Allocation: | **MTS-Security SIG meeting** -  |
| Relevant WI(s), or deliverable(s): |   |
|  |

**ABSTRACT:***A short discussion of the use of modelling and languages in making security systems, and technologies added to systems to provide security, both verifiable and testable.*

ETSI, primarily in TISPAN and to a lesser extent in OCG-Sec and MTS, has developed a number of guides on the development of security standards. The aim of these methods, summarised over the extracted figures below, is to ensure that security solutions in the standards domain meet the risk expectations for the context, and also to ensure that the solution is traceable from objective to requirement. The method is best shown in figure1.

The primary source is the "Threat, Vulnerability and Risk Analysis" (TVRA) method defined in TS 102 165-1 and more recently the guide to specifying requirements TR 187 011. In addition to these 2 guides ETSI has prepared guides giving patterns for security mechanisms in TS 102 165‑2 covering the structures of the CIA (Confidentiality, Integrity, Availability) paradigm.



Figure 1: Structure of security analysis and development in standards documents (from TS 102 165‑1)

The TVRA models a system consisting of assets. An asset may be physical, human or logical. **Assets** in the model may have **Weaknesses** that may be attacked by **Threats**. A **Threat** is enacted by a **Threat Agent**, and may lead to an **Unwanted Incident** breaking certain pre-defined security objectives. A **Vulnerability**, consistent with the definition given in ISO/IEC 13335, is modelled as the combination of a **Weakness** that can be exploited by one or more **Threats**. When applied, **Countermeasures** protect against **Threats** to **Vulnerabilities** and reduce the **Risk**. The relationships are shown diagrammatically in figure 2.



Figure 2: Generic security TVRA model from TS 102 165‑1

When we look at the major missing element it is in verification. This then requires us to ask what we want to verify/test. Security testing is not trivial – security breaches are increasingly multi-vectored and thus proving that any one protection measure works against its designed for attack does not give any guarantee that the asset is protected in general.

More recently TR 187 011 has identified the structure of security requirements.



Figure 3: Security objectives and requirements from TR 187 011

Security objectives should meet the criteria of being:

* realistic:
* The objective does not make unjustifiable demands on the target system. For example, in a secure environment it would be unrealistic to set an objective that all users should be able to view the secret passwords of all other users;
* achievable:
* it should be possible to meet the objective within the bounds of current or emerging technology without unreasonable cost;
* measurable:
* Once an objective has been met, it should be possible to view or otherwise validate its effect on the target system either directly or indirectly;
* relevant:
* the objective should be directly related to the general security of the target system and its environment;
* The objective should not detract from the overall purpose of the target system.

If a security objective is unable to meet all of these criteria, it should be revised or rejected.

# The role of modelling in security development

A key element of security analysis is that of identification of the Target of Evaluation. This requires modelling the interfaces exploited by the attacker, the modelling of assets in the system and their interaction, and the value of an attack (masquerade, manipulation, interception, etc.) to the attacker and its impact on the system and its legitimate users. The resulting detail system model is generally based on the published architectures, protocols and procedures relating to the context under study.

Modelling of the base system is therefore considered as essential to the viability of the risk analysis. The success of the risk analysis also requires to confirm the model with the developers of the base standard. If however the base standard has not been modelled itself in some easily accessible way the accuracy of the risk analysis, and the resulting countermeasures that it identifies as essential, will be compromised.

# The role for an extended TPLan

The current role of TPlan is as a notation for the construction of simple test objectives. This is achieved by the [paraphrased] structure of "When <*set of pre-requisites*> are met **ensure that** with this <*stimulus*> these <*post conditions*> are achieved". This is similar to the structure of requirements recommended in TR 187 011:

<Precondition><Stimulus><Response>

* The [optional] precondition indicates the circumstances or context that must be established before the requirement becomes valid.
* The stimulus defines the action which causes the security system to initiate a visible (measurable) response.
* The response defines the behaviour of the implementation on receiving the defined stimulus.

If we make the assertion that a Test Purpose is an objective (i.e. it is not a requirement), then by changing of some key-words the same notation can be adopted for writing objectives, and with some additional syntax can be developed as a testable language for writing syntactically correct objectives and requirements. Extending TPLan can therefore allow tool developers to incorporate syntax verification of requirements to the models.

NOTE: Enterprise Architect for example allows requirements models to be added to UML models but does not appear to follow a formalised language for the requirements part (as opposed to the UML from OMG).