# SECURITY MODELLING FOR RISK ANALYSIS

Lam-for Kwok[1] and Dennis Longley[2]
*[1]City University of Hong Kong, Department of Computer Science, Hong Kong, cslfkwok@cityu.edu.hk ;[2]Information Security Research Centre, Queensland University of Technology, Brisbane, Australia, d.longley@qut.edu.au*

**Abstract:** A security model to facilitate the recording and investigation of organisational security data is proposed; this model employs a directory structure for security entities and relationships. The model database with associated software may then be employed to develop and display organisational threat networks representing the risk environment of the organisational information processing and communication system. Thereafter the design of the defence systems may be facilitated by interactive procedures to determine appropriate countermeasure structures.

**Key words:** risk analysis, security standards, security models, security documentation, countermeasures, threat trees.

## 1. INTRODUCTION

The high cost of data collection for risk analysis projects has tended to favour high level methodologies focusing upon organizational structures; Baskerville[1] and Craft et. al[2] provide detailed accounts of such information security risk analysis methodologies. Nevertheless effective information security risk management for current highly complex systems depends upon a detailed knowledge of the system, and its environment. The Risk Data Repository (RDR)[3-7] was developed on the philosophy that risk analysis should be based upon a relatively simple and transparent analysis of comprehensive security data, rather than complex analyses of limited data. It aimed to store security data electronically so as to facilitate security studies and provide an effective means of developing and maintaining security

Y. Deswarte et al. (eds.), *Security and Protection in Information Processing Systems*

documentation. When combined with a technique for representing the effectiveness of countermeasures[8], it could form the basis of a security officer's workbench.

Experience with the RDR indicated the requirement for a uniform method of representation for the various security entities and relationships. This paper explores the application of a security model based upon the RDR concepts and a directory structure for entity representation. An electronic security database and supporting software, of the type described in this paper, would not only alleviate the task of security documentation development, it would also significantly reduce the effort of initial data collection and subsequent updating. Moreover a common form of security documentation would greatly facilitate the importation of external security expertise, from vendors, consultants, standards bodies etc. It would also support processes to maintain system security when disparate systems are inter-networked or merged.

# 2. SECURITY MODEL

## 2.1 Security Documentation

Security documentation is an essential component of organisational information security because it provides a common source of information to the wide range of staff with information security responsibilities, and minimises the duplication of data collection effort. Such documentation should contain a broad range of information including inter alia business, management, administrative, operational and environmental contexts.

However, the cost of developing and maintaining such documentation is high, and it is suggested that such cost and effort may be reduced if conventional documentation were replaced by a security information database with supporting software. Such database / software would not only significantly reduce the effort of data recording and retrieval, it could also assist the security officer in the analysis of risk scenarios, design and maintenance of information security defence systems etc.

The effective cost of the database/ software design could be minimised if it were widely deployed, which implies the need of a common security model to guide the design of the database. This paper describes a security model for that purpose. The first stage in the model development lies in the classification of the various entities relevant to organisational security (see 2.2).

## 2.2 Classification of Entities

The classification of security entities in the proposed model was discussed in a previous paper by Fung et al[9] and was based upon a directory model[10]. The wide array of entities relevant to organisational security are grouped within classes, and uniquely identified, in the model. It is possible to generate a macro view of the system security by commencing the data collection with high level entities: major sites, large IT and communication systems, major applications and user groups etc. and then refine this view as detail of the component entities are recorded.

The proposed model uses but does not at this stage prescribe a directory structure. It simply suggests that an entity:

- has a unique identifier (an object identifier) indicating its position within its family grouping;
- has an arbitrary set of attributes defined by (Tag Value) tuples;
- can be linked with other entities, and such linkages are themselves entities of the model.

| Description | OI | Description | OI | Description | OI |
|---|---|---|---|---|---|
| **Systems** | 1 | Hardware | 1.1 | | |
| | | Software | 1.2 | | |
| | | Platforms | 1.3 | | |
| | | Networks | 1.4 | | |
| | | Applications | 1.5 | | |
| | | Users | 1.6 | | |
| | | Assets | 1.7 | | |
| **Environment** | 2 | Locations | 2.1 | Sites | 2.1.1 |
| | | Services | 2.2 | Power | 2.2.1 |
| | | | | Communications | 2.2.2 |
| | | | | Water/Drainage | 2.2.3 |
| **Security** | 3 | Threats | 3.1 | Threat Types | 3.1.1 |
| | | | | Threat Trees | 3.1.2 |
| | | Defences | 3.2 | Countermeasures | 3.2.1 |
| | | | | Threat Countermeasure Diagrams | 3.2.2 |
| **Procedures** | 4 | Internal | 4.1 | Policies & Procedures | 4.1.1 |
| | | | | Standards | 4.1.2 |
| | | External | 4.2 | | |
| **Relationships** | 5 | Link Type 1 | 5.1 | | |
| | | Link Type 2 | 5.2 | | |
| | | Link Type 3 | 5.3 | | |
| | | Link Type 4 | 5.4 | | |

*Figure 1.* Directory Classification of Entities
(OI = Object Identifier is a unique numerical identifier for each entity)

A proposed directory system is illustrated in Fig. 1. Within this scheme every entity is given a unique identifier (termed object identifier OI)

according to its position in the directory[9]; hence such an identifier may be allocated to an IT system, network, file server, room, class of users or a paragraph in the standard documentation. This classification scheme provides for groupings of entities within classes and hence facilitates a top down approach to the documentation.

## 2.3 Attributes, and Relationships between Entities

The classification scheme described above (see 2.2) provides a list of entities. Additional information about particular entities (e.g. the protocol of a network) can be given as an attribute of that entity (Tag = Protocol, Value = TCP/IP). Hence security relevant attributes may be allocated to entities as the need arises from the model.

Information processing systems involve inter-relationships between their various entities, e.g. workstations are connected to networks. These inter-relationships are themselves entities within the scheme (see Fig. 1). Relationships thus have object identifiers; a particular relationship is given a unique identifier that is itself, a child of a relationship type identifier. Hence if the model records that a given PC (with object identifier PC_OI) is *connected* to a specific LAN (LAN_OI) then this fact is recorded with a Relationship OI (Connect_OI.1) belonging to the Relationship family Connect (Connect_OI).

## 2.4 Developing the Model

The proposed scheme is capable of describing security scenarios of complex systems down to any required level of detail, but selecting the type of information to be collected, and the requisite level of detail, is no mean task. It is suggested that the process commences with a top level model, explores the risk environment at this macro level and use the results to guide the refinement of the model.

Hence the Systems Entity contains the class Platforms used to represent major organisational systems. A simple security macro model can then be developed by identifying the major security facets of a platform:

- its location, i.e. identifying the building or site hosting the platform and linking them with relationship of type – *located;*
- the classes of *assets* processed by the platform, i.e. linked to the platform with the relationship type – *processed by;*
- the external services required to ensure the availability of the platform.

Once a macro model has been developed the security features of the platform may then be explored, using the techniques described in the next sections (see 3 and 4). These studies using the interactive tools described

below will then provide guidance on the next stage of the model environment.

# 3. SECURITY MODELLING

## 3.1 Overview

Information security officers are concerned with threat events having the potential to damage organisational information assets and systems. They need to predict and prioritise such threat events, in order to develop and maintain cost effective security measures.

A threat tree is such a tool for risk analysis, since it displays the potential set of outcomes for a given threat event. If all the potential threats are considered then a forest of threat trees can be developed, the upper level comprising the initial threats, the leaves the potential outcomes. The trees are not necessarily disjoint, since some of the tree nodes may be common to a number of trees, e.g. physical damage to computer equipment is likely to arise from fire, flood, malicious damage etc. Hence the collection of trees may actually represent a network, with intrinsic threats at the top and the undesirable outcomes at the bottom (see Fig. 6).

Having identified and prioritised the threat scenarios in terms of the probabilities and impact severities, the security officer has the task of selecting, installing, customising and maintaining appropriate security countermeasures. The role of a countermeasure may be considered to be that of cutting, or at least weakening, a branch of the threat network so as to inhibit one or more undesirable outcomes. Countermeasures themselves are physical devices, administrative procedures etc. that may be rendered ineffective or bypassed, and should often themselves be protected by supplementary countermeasures.

## 3.2 Threat Propagation

### 3.2.1 Threat Entity Relationships

The proposed security directory system recognises threats classified from top levels (see Fig. 2). Each of these threats may be assigned object identifiers (OI) according to their position in the hierarchy.

Threats become significant to an organisation when they impact upon a System or Environmental Entity, e.g. *Malicious Physical Damage to Finance LAN router*, *Accidental Physical Damage to a HQ Site Power Supply, Fire in I.T. Building*. Like Threats, each of these target System or Physical Environment entities is allocated an object identifier in the model.

Thus a threat event represents a threat impacting upon some specific entity, and can be represented by a link between the threat and the impacted entity OIs (see Fig. 3). The threat events are represented as nodes in the Relationship directory, under the class Threat_Entity: i.e.

- TE is a linkage (Threat OI, Entity OI).
- These linkages are of the class Threat Entity (TE) and each individual TE is allocated an OI in the model.

| | | |
|---|---|---|
| Threats (3.1.1) | Physical (3.1.1.1) | Natural (3.1.1.1.1) |
| | | Malicious (3.1.1.1.2) |
| | Environmental (3.1.1.2) | Fire (3.1.1.2.1) |
| | | Flood (3.1.1.2.2) |
| | | Earthquake/ Volcanic Eruption (3.1.1.2.3) |
| | | Severe Weather (3.1.1.2.4) |
| | | Asteroid Impact (3.1.1.2.5) |
| | Personnel (3.1.1.3) | ..... |
| | Network (3.1.1.4) | ..... |

*Figure 2.* Classification of Threats

### 3.2.2 TE – TE (TETE) Relationships

Threats propagate, for example:

- as fire through geographically related physical environments;
- by damaging essential services such as power supplies causing systems to shut down;
- by interactions among information processing/ communication systems.

A fire in a room may cause physical damage to equipment located there. In effect an *Incident Threat* impacting upon an *Incident Entity* may cause a *Target Threat* to impact upon a *Target Entity*. In the model this threat propagation is represented a link between two threat events or TEs (see Fig. 4):

- Incident TE: Link (IncTE_OI) between Incident Threat (IncT_OI) and Incident Entity (IncE_OI);
- Target TE : Link (TarTE_OI) between Target Threat (TarT_OI) and Target Entity (TarE_OI);
- TETE : Link (TETE_OI) between IncTE_OI and TarTE_OI.

Threat propagation is not inevitable, e.g. there are a number of factors determining if a given piece of equipment will be damaged by fire in a room: size of the room, flammability of materials in the room, fireproofing of the equipment etc. Hence a TETE may have an attribute termed Vulnerability Index (0 <= VI <=1) to indicate the probability of the threat propagation.

### 3.2.3 Threat Tree Development

Threat trees (see Fig. 5) may be developed automatically in the model. Hence there is a Threat Tree Development Algorithm:

1. Select the initial threat event (Root TE).
2. Find a TETE with Root TE as its Incident TE.
3. Get the corresponding Target TE (second component of TETE link)
4. Target TE is next node in Threat Tree.
5. Repeat 2 until there are no more candidate TETEs
6. Go to next child node in the Threat Tree and set this as Root TE
7. Repeat 1 until no more child nodes.

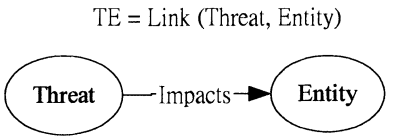


*Figure 3.* TEs Represent Threat Events

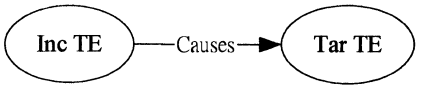


*Figure 4.* TETEs Represent Threat Event Propagation

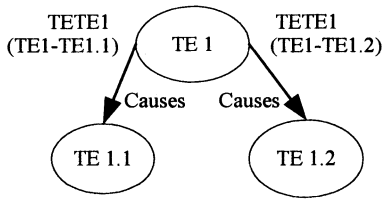


*Figure 5.* TETEs Provide the Links in a Threat Tree

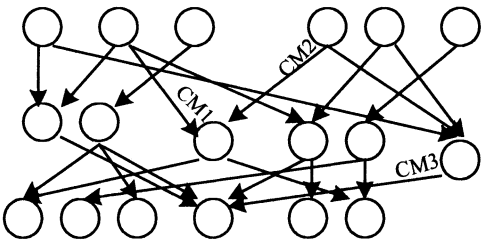


*Figure 6.* Threat Networks including Countermeasure

The Threat Tree nodes may be assigned Object Identifiers and hence the Threat Trees are themselves are represented within the model. The automatic development of threat trees described above, however, involves a major manual effort in the production of TEs and linkages between TEs i.e.

TETEs. These TETEs represent security knowledge on threat propagation; the potential number of TETEs could be in the billions for a reasonable sized organisation. Fortunately the model provides a solution to this problem, since the proposed directory approach organises security entities in classes. Multiple individual TETEs can thus be replaced by individual generic TETEs containing details of entity classes (see 3.2.4).

### 3.2.4 Generic TETEs

#### 3.2.4.1 Incident and Target Entity Linking

The directory classification of security entities implies that simple relationships exist between common entities. Hence locations will have object identifiers that reflect the geographical relationships (see Fig. 7).

<table>
<tr><td rowspan="6">Physical Environent / Location / Site (OI = PELS)</td><td rowspan="4">HQ Site (PELS.1)</td><td rowspan="3">Admin Building (PELS1.1)</td><td rowspan="2">Ground Floor (PELS1.1.1)</td><td>Room 1 PELS (1.1.1.1)</td></tr>
<tr><td>Room 2 PELS (1.1.1.2)</td></tr>
<tr><td colspan="2">.....</td></tr>
<tr><td>1st Floor (PELS 1.1.2)</td><td></td></tr>
<tr><td colspan="2">Manufacturing Building (PELS.1.2)</td><td></td><td rowspan="2"></td></tr>
<tr><td>Branch Site (PELS.2)</td><td></td><td></td></tr>
</table>

*Figure 7.* Object Identifiers for Locations

The generic TETE for fire propagation recognises that a fire at HQ Site can spread to the Admin Building, which in turn can spread to floors in that building and from the floors to rooms on those floors. Hence the generic TETE for fire propagation can be expressed in terms of wild card OIs indicating parent and child relationships of the Incident and Target Entities (i.e. sites, buildings, floors and rooms etc.)

In general the conditional relationships between the Incident and Target Entities may be more complex. For example a TETE may take the form:

*Fire in a Physical Location causes Physical Damage to Hardware on condition that:- Hardware is Located in the Physical Location.*

The TETE thus needs to store the Incident and Target Entities as wild card representing Physical Location and Hardware entities respectively. In addition the TETE stores the information regarding the required link between the Incident and Target Entities, as an attribute of the TETE. The use of object entities to identify Relationship classes facilitates the storage of this attribute information, e.g. *Located_In.* With more complex conditions

between the TETE Incident and Target Entities, a Linkage Condition Table may be included as an attribute of the TETE.

A generic TETE may thus replace a multiple of specific TETEs, and such generic TETEs allows security knowledge to be directly imported to the model. The threat tree development now involves a search for specific Target Entities that:

- match a TETE Target Entity class, and
- satisfy the Incident – Target Entity relationships as specified in Linkage Condition Table.

#### 3.2.4.2 Vulnerability Indices (VI).

The probability associated with a specific threat propagation will naturally depend upon the particular threats and entities, and their attributes; e.g. fire propagation will have a higher probability for wooden buildings. Hence VI Condition Tables may also be stored as TETE attributes, to facilitate the estimation of probability associated with a threat propagation between specific entities.

### 3.2.5 Role of Threat Networks

#### 3.2.5.1 Development of Threat Networks.

Threat Trees have a single threat event (TE) root; these trees may contain duplicate nodes implying that such nodes have more than one parent. Combining trees with various threat event roots may also result in the combination of duplicate intermediate nodes. Hence the total set of threat scenarios will normally take the form of a network (see Fig. 6). Duplicate nodes are significant in the interpretation of Threat Networks and in the design of subsequent defence systems.

#### 3.2.5.2 Outcome Probabilities.

It is extremely difficult to assign accurate probabilities to a plethora of potential information security events. The way forward, proposed in this model, is to allow for the inclusion of probabilities as transparently as possible, and to suggest the provision of interactive tools to facilitate the task of probability estimation, i.e. the estimator should at least be given the opportunity to explore the sensitivity of the outcome to the estimate.

A Threat Tree provides for a comparatively straightforward estimate of outcome probabilities based upon:

- the probability that the root threat event will occur; and
- the probability of threat propagation (VI) for each link between the root and the outcome.

The probability of an outcome, arising from a threat event, will decrease with the number of serial links in the path, and increase with the number of parallel paths. Given two parallel paths of different lengths, the outcome probability will tend to be more sensitive to the VIs of the links in the shorter path.

An interactive tool that displays the variation of outcome probabilities with variations in the VI of a selected link, would allow a security officer to focus on the links with the highest impact on outcomes and hence prioritise the effort in estimating probabilities.

#### 3.2.5.3 Outcome Impacts.

An outcome in the Threat Networks (see Fig. 6) represents a threat to an organisational asset, e.g. loss of confidentiality of client account data. The previous sections have alluded to security officers prioritising their efforts towards outcomes with the largest impacts, but so far have not discussed the estimation of such outcome impacts. This task is onerous, particularly if there is a requirement to assign quantitative measures to such impacts.

The fundamental problem lies in the relationship between a technical event, such as compromise to a business dataset, and the subsequent impact upon the organisation. Security officers would normally experience significant difficulty in trying to obtain a host of such quantitative measures from business management, let alone maintaining the validity of such data as organisational environments evolve. This aspect of risk analysis was studied in the RDR development[6, 11]. The approach suggested was:

- Defer the allocation of the quantitative measure and assign textual statements to outcomes, e.g. *illicit disclosure of this asset could lead to major loss of client confidence.*
- Deduce impacts from information on impacts associated related assets.

The allocation of impact statement to assets can still prove to be an excessive task for organisations with large variety of datasets. Anderson[11] demonstrated that given knowledge of the inter-relationship of data items, it was possible to deduce impact statements for a set of data, from explicit impact statements given for a related dataset. The proposed security model facilitates such imputation of impacts.

### 3.2.6 Automatic and Interactive Threat Network Development

The concept of automatic Threat Network development sounds attractive but it depends upon the initial storage of a significant amount of general and local security knowledge. If the requisite generic TETEs are not stored in the model then the search will end prematurely and a potential threat path will be omitted, giving a false sense of security. Moreover, even if the requisite

TETEs are stored, the path will still end prematurely if local entities, or local relationships between those entities, are not included in the model database. As ever, the model results are only as good as the model itself.

However, the proposed directory structure does provide two ways forward. Firstly, given a common directory structure then generic TETEs may be imported into a local model. Hence a large organisation can effectively export its security expertise from head office to local branches. Moreover a study of the Incident and Target Entity types and conditional relationships for these imported TETEs provides a clue to the type of entity and relationship that should be included in the local model.

Secondly the proposed model provides guidance on the development of the local security database when Threat Tree Networks are developed in an automatic / interactive manner. Consider a node in the Threat Network with a premature end node in a path. An interactive session can report the class of Target Entity, and Incident – Target relationships for each TETE with an Incident Threat event matching the node; for example, the interactive session may suggest a check for equipment located in a room vulnerable to flood.

# 4. DEFENCE MEASURES

## 4.1 Overview

The previous section discussed the role of the model in providing a security officer with an insight into the organisational threat scenario. Threat Networks display the threat paths to organisational assets and Vulnerability Indices (VI)/ impact statements prioritise potential outcomes. The security officer then has the task of prioritising defence systems according to the system risk, expressed in terms of outcome impacts and probabilities.

The security officer is thus responsible for the design, implementation and operation of cost effective defence measures designed to minimise the identified system risk. This section describes how the proposed model may be used in this role.

## 4.2 Countermeasures

A study of Threat Networks can identify branches that have the effect of increasing the probability of undesirable outcomes. Countermeasures are employed to minimise the probability of a threat propagation leading to such

unacceptable outcomes. Hence a study of Threat Network branches may assist in the optimum deployment of countermeasure (see Fig. 6)

Reducing the probability of a threat propagation is equivalent to reducing the effective VI of a TETE (see Fig. 8); e.g. if an attacker logs onto a computer system there is a moderate to high probability that a sensitive file stored on the system will be compromised with loss of confidentiality. A password access control system protecting the file will mitigate against such an event. Hence in Fig. 8,

- Incident TE: Attacker Accesses – File Server,
- Target TE: Loss of Confidentiality – Personnel File,
- TETE: (Incident TE , Target TE),
- TETE Condition: Personnel File STORED ON File Server
- Countermeasure: Password Access Control.

Countermeasures are included as Security Measures in the Directory scheme. An earlier publication on countermeasures[8] discussed a model highlighting the role and effectiveness of countermeasures. The effectiveness of any countermeasure depends upon its inherent components, and threats to those components can seriously affect the countermeasure performance. For example, a firewall is highly dependent upon the effectiveness of its rules. A logical attack may allow malicious packets to satisfy inadequately formulated firewall rules for access (i.e. bypassing the countermeasure), whilst unauthorised physical access to the firewall hardware could allow the attacker to modify the firewall rules, rendering the firewall ineffective.

Countermeasure structures (see Fig. 9) may thus be modelled in terms of:

- the incident TETE, i.e. the threat propagation to be countered;
- the components of the countermeasure, i.e. those aspects of the countermeasure that determine its effectiveness; and
- the residual TETEs, that could impact upon those components, and hence compromise the operation of the countermeasure.

The residual TETEs represent potential attacks on the countermeasure components, e.g. *Illicit Access to Firewall Equipment* CAUSES *Illicit Modification to Firewall Rules.*

The security officer has the role of protecting countermeasures against attacks on its components, and may therefore deploy supplementary countermeasures to ensure the effectiveness of the original countermeasure (see Fig. 10), e.g. an intrusion detection system may be employed to counter sophisticated attacks bypassing firewall rules. The supplementary countermeasures may also be implemented in the form of procedural security, e.g. an access control system may have a residual TETE: *Attacker Guesses Password Causes Attacker to Gain Access to System* and this could be countered by an organisational procedure mandating strong passwords.

## 4.3 Threat Countermeasure Diagrams (TCDs)

TCDs are used to record the complete structure of major and supplementary countermeasures (see Fig. 10). A TCD is a tree of countermeasure structures with the prime countermeasure as the root and the supplementary countermeasures as the child nodes. The TCD records the rationale of a countermeasure system. Given that the supplementary countermeasures will themselves have vulnerable components, and hence residual TETEs, it would appear that TCDs will grow indefinitely. The decision to employ supplementary countermeasures depends upon their role in the effectiveness of the root countermeasure and this aspect of TCDs is discussed in 4.5.

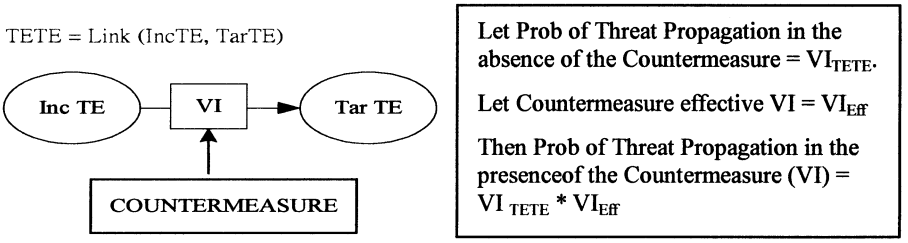


*Figure 8.* The Role of the Countermeasure is to reduce the Probability of the Threat Propagation, (VI) of a Threat Network Link (TETE)

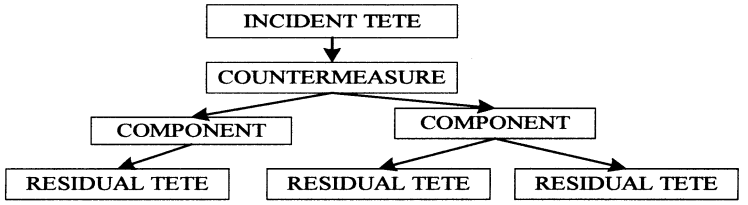


*Figure 9.* Countermeasure Structure

## 4.4 Design of Defence Systems

The countermeasure structures (see Fig. 9) and TCDs (see Fig. 10) fit well into the proposed directory structure. The various facets of the countermeasure (incident TETE, components and residual TETEs) may be allocated object identifiers under the countermeasure entity, and the nodes of a TCD may be treated in a similar manner to those of Threat Trees. These defence structures represent significant security knowledge. The

countermeasure structure provides details of the countermeasure vulnerabilities; the corresponding TCDs give advice on the secure installation of the countermeasure. Hence generic countermeasure structures and corresponding TETEs may be imported into the model. The defence structures may be developed in a similar manner to the automatic development of Threat Trees.

The first stage in the development of a defence system lies in the construction of generic countermeasure structures (see Fig. 9). The security designer with detailed knowledge of the countermeasure determines the incident TETEs, essential components, residual TETEs etc. These incident and residual TETEs are developed as *generic* TETEs (see 3.2.4), i.e. the incident and target entities contained in the TETEs refer to a class of systems or physical environment entities. These constructs are imported into a local organisational information security model.

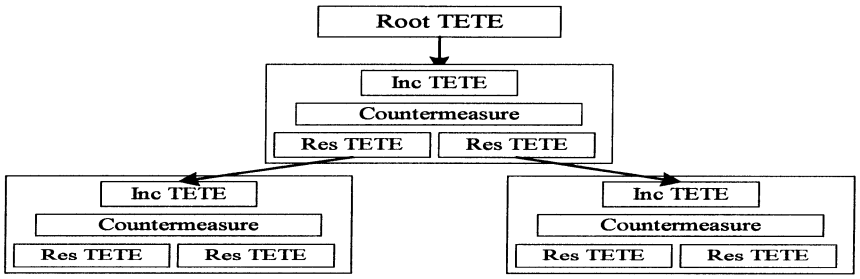


*Figure 10.* A Threat Countermeasure Diagram (TCD) displays Supplementary Countermeasures maintaining the effectiveness of the Root Countermeasure

A security officer now selects a link in a Threat Network for the placement of a countermeasure. This link represents a local TETE, since it contains details of the organisational actual entities, and now becomes the root TETE of a TCD. An automatic search amongst the generic countermeasure structures produces one or more countermeasures with incident generic TETEs matching the TCD root TETE. The security officer interactively selects one of the offered countermeasures and the first node of the TCD is inserted. The incident generic TETE is instantiated to the local entities, the residual TETEs are correspondingly instantiated.

The security officer may then proceed in a similar manner to the selection of supplementary countermeasures, countering the residual TETEs.

Alternatively complete generic TCDs may be imported and then instantiated by the entities of the root TETE. In effect the warnings and recommendations contained in security information brochures can be directly imported into the model.

There remains the problem of the depth of supplementary countermeasures. Currently these decisions are made, often implicitly, on the basis of experience and judgment. The model can assist with this aspect of countermeasure design by consideration of the VIs associated with the incident and residual TETEs in the total structure (see 4.5).

When countermeasure systems have been selected and implemented they become part of the total information security model and new equipment and procedures will be added. This raises the question, can the countermeasures introduce new threats, i.e. additional paths in the Threat Network? There is plenty of anecdotal evidence of water damage caused by fire extinguisher sprinklers. Thus the organisational Threat Network should be redeveloped on the updated model to test for such eventualities.

## 4.5 Defence Effectiveness

The role of a countermeasure is to reduce the intrinsic probability of threat propagation, i.e. reduce the VI of the incident TETE (see 4.2); the model allocates an effective VI ($VI_{eff}$) to the countermeasure (see Fig. 8). The countermeasure $VI_{eff}$ depends upon the VIs of its residual TETEs. Consider a simple password system, and assume that the only means of rendering the system ineffective is to correctly guess a PIN. The residual TETE may take the form *Attacker Guesses PIN Causes Access System Compromise*. In this case the VI of the residual TETE (0.0001) is the countermeasure $VI_{eff}$. If there are *n* residual TETEs, then the worst scenario is that each potential attack on the countermeasure is simultaneously and independently undertaken; the countermeasure $VI_{eff}$ is then given by

$$VI_{eff} = 1 - (1 - VI_1) * (1 - VI_2) *... *... (1 - VI_n)$$

If any residual TETE has a VI close to 1, then $VI_{eff}$ is also close to 1 and the countermeasure is ineffectual. If a supplementary countermeasure guards against a high residual TETE then that residual TETE value is reduced by a factor equal to the $VI_{eff}$ of the supplementary countermeasure. Tools may be developed to compute countermeasure effectiveness interactively based upon estimates of VIs of residual TETEs. A simple grading scheme of *high, neutral and low VI* can be easily incorporated to assist in the decisions on supplementary countermeasures.

## 5. CONCLUSIONS

The proposed model seeks to provide a methodology to assist in the problem of maintaining information security within complex organisational environments. It does not claim to be a simple solution to a complex problem. It does, however, claim to provide a methodology for information security defence design by focusing organisationally available skills and expertise to the local problem, making the most effective use of available system data, guiding the collection and recording of additional security data.

The basis of the methodology is a database of security entities with associated tools, used interactively to develop security insight of the modelled system, which in turn guides the collection and recording of additional security data so that the model increasingly reflects the organisational security environment. The model facilitates the distribution of security expertise and experience to operational environments. Security officers of a large organisation will now have common tools to capture local data, develop security models for local systems and use those models to improve the local security scenario.

To date experience with the model has been limited to the development of prototypes to test the concepts of Threat Tree and TCD development using the procedures described in the paper. The development of a user friendly package with graphical displays is due to commence shortly.

## ACKNOWLEDGMENTS

The authors acknowledge the valuable help, advice and suggestions received from colleagues and students in the development of the ideas that lead to the production of the model; in particular, contributions from Prof W. Caelli, Dr A. Anderson, Dr A. Tickle, Ms P. Fung and Mr M. Branagan.

## REFERENCES

1 R. Baskerville Information Systems Security Design Methods: Implications for Information Systems Development. *ACM Computer Surveys*, **25**(4) 1993, 375-414.

2 R. Craft, G. Wysss, R. Vandewart and D. Funkhouser. An Open Framework for Risk Management. *21st National Information Systems Security Conf.*, 6-9 Oct. 1998, Crystal City, Virginia, USA.

3 L. F. Kwok and D. Longley. Code of Practice: A Standard for Information Security Management. *Information Security in Research and Business, Proc. IFIP TC11 13th Int. Conf. on Information Security* (Editors: Yngstrom and Carlsen), IFIP Sec.'97, Copenhagen, Denmark, 14-16 May 1997, Chapman & Hall 1997, pp.78-90.

4 L. F. Kwok. A Hypertext Information Security Model for Organisations. *Information Management and Computer Security*, **5** (4) 1997, 138-148.

5 L. F. Kwok and D. Longley. Information Security Management and Modelling, *Information Management and Computer Security*, 7 (1) 1999, 30-39.

6 A. Anderson, L. F. Kwok and D. Longley. Security Modelling for Organisations. *Proc. 2nd ACM Conf. on Computer and Communications Security*, CCS'94, Fairfax, Virginia, USA, 2-4 Nov 1994, ACM Press 1994, pp.241-250.

7 L. F. Kwok and D. Longley.. A Security Officer's Workbench. *Computers and Security,* **15** (8) 1996, 695-700.

8 W. Caelli, D. Longley, and A. B. Tickle. A Methodology for Describing Information and Physical Security Architectures. *IT Security: The Need for International Cooperation, Proc. IFIP TC11 8th Int. Conf. on Information Security* (Editors Gable and Caelli), IFIP Sec.'92, Singapore, 27-29 May 1992, NY:Elsevier Science Publishers 1992, pp.277-296.

9 P. Fung, L. F. Kwok and D. Longley. Electronic Information Security Documentation. *ACSW Frontiers 2003* (Eds. Johnson, Montague and Steketee), Australasian Information Security Workshop (AISW2003), 4-7 Feb. 2003, Adelaide, Australia, pp25-31.

10 *The Directory, CCITT Rec. X.500-X.521* | ISO/IEC Standard 9594:1993.

11 A. Anderson. The Object Oriented Modelling of Information Systems SecurityRisk, *PhD thesis*, Queensland University of Technology 1997.