A Novel Risk Assessment and Optimisation Model for Multi-objective Network Security Countermeasure Selection Problem

Valentina Viduto[a,*], Carsten Maple[a], Wei Huang[a], David López-Peréz[b]

[a] *Institute for Research in Applicable Computing, University of Bedfordshire, Park Square, Luton, Bedfordshire, LU1 3JU, United Kingdom*
[b] *Centre for Telecommunications Research, King's College London, Strand, WC2R 2LS, United Kingdom*

Abstract

Budget cuts and high demand in strengthening the security of computer systems and services today are problematically balanced facts. Poor system knowledge and inappropriate selection of security measures may lead to unexpected financial and data losses. This paper proposes a novel risk assessment and optimisation model (RAOM) which can be used as an extension of a standard risk assessment procedure to represent a multi-objective security countermeasure selection problem, the purpose of which is to minimises the risk presented by network vulnerabilities in relation to financial investments. A multi-objective Tabu Search (MOTS) algorithm has been developed to construct an efficient frontier of non-dominated solutions and has been compared to an Exhaustive Search (ES). It is discovered that the proposed approach provides near optimal results for this type of the problem and can provide a model used to support financial investment decision

[☆] This Research is financially supported by EPSRC, whose support is very appreciated
[*] Corresponding author.Tel.:+44(0)1582 742159
Email addresses: valentina.viduto@beds.ac.uk (Valentina Viduto), carsten.maple@beds.ac.uk (Carsten Maple), wei.huang@beds.ac.uk (Wei Huang), david.lopez@kcl.ac.uk (David López-Peréz)

making.

Keywords: Financial decision support, Risk assessment, Multi-objective optimization, Tabu Search.

---

1. Introduction

In the IT sector, most organisations implement security standards to be competitive and trustworthy parties that run highly integrated and secure businesses [1]. However, despite regulations, law and awareness of security measures, data breaches continue to grow and evolve. For instance, according to recent surveys, by around 84% of UK organisations suffered at least one data breach in 2007 [2].

When an organisation performs regular risk assessments of assets and services, the risk of experiencing a data breach may decrease. However, frequently, decisions on what security measures should be implemented are made based on the personal experience of the decision maker, who is often unaware of specific system weaknesses and new threats. In order to solve this issue, researchers have proposed a number of models relating to qualitative and quantitative risk assessment approaches, where attack trees and attack graphs are used to estimate the shortest attack paths and related security costs [3, 4, 5, 6, 7]. However, these models lack of practical sense and cannot support cost-effective security decisions.

A cost-effective and coherent risk assessment should study the relationships among system vulnerabilities, threats and countermeasures. Knowing potential risks gives the ability to organisations to make effective decisions on what security countermeasures should be implemented before any potential threat can successfully exploit system vulnerabilities. NIST SP800-30,

ISO 27001 and ISO 17799 are common methodologies, which provide some guidelines on how risk should be assessed and how countermeasures should be selected [8, 9, 10]. However, these guidelines do not provide a specific method for assessing risk related factors, i.e., vulnerabilities, threats, and addressing them via security countermeasures.

Security countermeasure selection problems have received a great deal of attention in the recent literature [11, 12, 13, 1]. However, existing approaches deal with this problem from very different perspectives. The authors in [12] analyse the countermeasure selection in relation to residual vulnerabilities, which are represented as uncovered existing vulnerabilities. The idea behind their approach is to maximise the coverage of existing vulnerabilities by implementing the selected set of countermeasures, thus, to minimise the residual vulnerability (uncovered). Another approach to select a portfolio of countermeasures in relation to investment costs is by analysing the residual damage in the system if a system hole is not fixed [13] and considering a set of controls in a form of disabling, enabling or patching a service or application.

Although they are very detailed in some aspects, current countermeasure selection approaches miss some other details. For example, applying a countermeasure may eliminate some risks, but generate new ones under certain circumstances. Therefore, it is not enough performing risk assessments and independently selecting security countermeasures, but it is necessary to understand the bi-directional relationship between them both. In this way, we can make sure in a cost-effective manner that organisations are aware of possible data losses and that the adequate security countermeasures are in place.

Due to the lack of studies on this topic, this paper investigates risk

assessment methodologies and provides a tool to select security countermeasures taking financial costs and residual risks into account. More specifically, based on NIST SP800-30 guidelines, we propose a risk assessment and optimisation model (RAOM) to satisfy organisational security needs in a cost-effective manner, systematically present our security countermeasure selection problem and formulate it as a multi-objective optimisation problem, where variables such as financial cost and residual risks may affect the final solution. We also propose a tailored Tabu Search(TS)-based heuristic approach to solve the proposed multi-objective optimisation problem and asses the qualities of its solutions with respect to optimal ones.

2. Risk Assessment and Optimisation Model (RAOM)

In this section we present our risk assessment model, compare it to NIST SP800-30 framework and formally define our multi-objective optimisation problem.

ROAM consists of two processes, risk assessment and optimisation routine. The purpose of the proposed RAOM is to provide the foundation of an effective risk assessment procedure, containing practical methods necessary for assessing risks and cost-effectively minimising them through security countermeasures. Figure 1 illustrates the flow chart of RAOM, which describes firstly the risk assessment stages (Part A), including identification of risks through a vulnerability assessment, their impacts through the analysis of threats mapped onto vulnerabilities and secondly, optimisation routine (Part B), which is used to find optimal solutions in a cost-effective manner.

The first stage in our proposed risk assessment procedure is to identify essential organisation's systems and functions, which cannot be interrupted

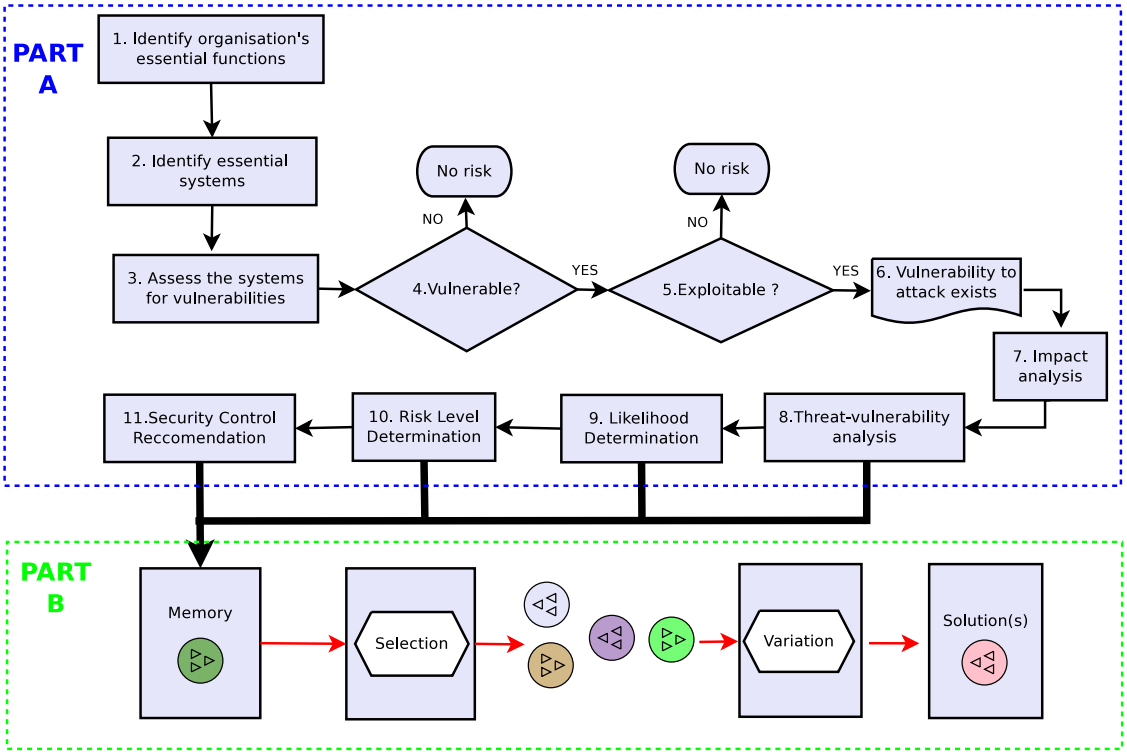


Figure 1: Risk assessment and optimisation model (RAOM) flow chart

under any circumstances. Then, these systems and functions are assessed for vulnerabilities, because if there is a vulnerability in the network, there is the risk that a threat exploits the vulnerability, and hence the organisation may face unexpected technical damages and financial expenditures. Vulnerabilities, technical or nontechnical, can be identified in four ways: using automated vulnerability scanning tools, performing penetration tests on systems, using vulnerability modelling techniques and assessing previous risk assessment IT documentation. Once the vulnerabilities are characterised, it is important to identify the threats that can exploit them. Vulnerabilities can only be translated into risk if there is a threat able to exploit them. If

we can estimate vulnerabilities and threats, we can derive the level of risk in an organisation. Our aim is then to reduce this level of risk by selecting the appropriate set of countermeasures in a cost-effective manner.

2.1. Definition of Vulnerabilities

Vulnerabilities are the weaknesses or flaws in system security procedures, design or internal controls that can be triggered or intentionally exploited, resulting in a security breach or a violation of security policy. The National Vulnerability Database provides a source of technical vulnerabilities [14]. Every vulnerability may have a different impact level on a system.Traditionally, the impacts of a vulnerability on confidentiality, integrity and availability (CIA) are considered, where for each one of them, there are three impact levels: partial (P), complete (C) and none (N). The tuple comprised of the three potential impact levels on CIA of a vulnerability is then translated into an impact sub-score I_{sub} with range [1,10], where 1 is the lowest and 10 is the highest impact. I_{sub} can also be retrieved from the National Vulnerability Database [14], and depends on the inherent characteristics of the vulnerability: exploit range, attack complexity, level of authentication needed.

Table 1 provides an example of the potential impact on CIA and sub-score I_{sub} of some vulnerabilities. Furthermore, each vulnerability impact sub-score I_{sub} is mapped onto impact $I \in \{10, 50, 100\}$ where 10 indicates a low impact on CIA, 50 - medium, 100 - high. Impact value I has been introduced to scale variations of I_{sub}.

Let each vulnerability be represented as a single bit in the vulnerability vector:

Representation (Repr.)	Vulnerability	CVE number	Impact on CIA	Impact Sub-score
V_1	Default, missing or blank local user password	1999-0504	PPP	6
V_2	VirusScan NT 4.0.2 doesn not modify scan.dat file	1999-1195	PPP	6
V_3	Administrator password disclosure	2006-0561	CCC	10
V_4	IE version 5.01.5.5 and 6.0	2003-0344	PPP	6
V_5	SSH v1 in OpenSSH has various weaknesses	2001-0572	PPP	6
V_6	Cisco CSS11000 malformed UDP packet vulnerability	2004-0352	NNP	2
V_7	mysqld in MySQL 3.21 stores passwords in log file	1999-1188	PPP	6
V_8	MySQL 3.21 allows mysql users to gain root privileges	2003-0150	CCC	10
V_9	Execute arbitrary commands in wu-ftpd 2.6.1	2001-0550	PPP	6
V_{10}	wu-ftpd 2.6.1 with the restricted-gid option enabled allows local users to bypass access restrictions	2004-0148	CCC	10

Table 1: Vulnerability and corresponding CVE, impact information

$$\vec{V} = \{V_i\} = \{1, 0\} \, \forall i, \;\; i = 1, 2, ..., n. \tag{1}$$

where V_i represents an individual vulnerability. The value 1 indicates the presence of this vulnerability in the information systems, otherwise 0.

Moreover, as discussed before, impact on CIA sub-score I_{sub} of vulnerability i is mapped onto impact I_i as follows

$$I_i = \begin{cases} Low(10) & \text{when } 0.0 \leq I_{sub} \leq 3.9; \\ Medium(50) & \text{when } 4.0 \leq I_{sub} \leq 6.9; \\ High(100) & \text{when } 7.0 \leq I_{sub} \leq 10; \end{cases}$$

Because every vulnerability identified in the network is a potential risk, we include all vulnerabilities into our analysis. For example, if we assume that an organisation has identified 10 vulnerabilities, we set n to 10 and $V_i = 1 \ \forall n$.

2.2. Threat Analysis

The next step in the model is to perform a threat analysis, which consists of identifying potential threat sources and actions that may exploit system vulnerabilities. An attack can be defined as the action, in which a threat exploits a vulnerability that may create some risk in the system.

Information about threats can be gathered from the organisation's historical data base about the attacks recorded in system log files or by using threat modelling techniques, which can predict threats not known to the organisation. For example, modelling techniques, such as attack graphs, attack trees or an onion skin model [15, 16] have been used to predict new threats in particular scenarios.

Let each threat be represented as a single bit in the threat vector:

$$\vec{T} = \{T_j\} = \{0, 1\} \, \forall j, \ \ j = 1, 2, ..., m. \tag{2}$$

where T_j represents an individual threat. The value 1 indicates the presence of this threat in the information systems and otherwise 0.

Thereafter, based on data breaches reports, logged attack attempts and self-expertise, we can match threats to vulnerabilities (Table 2) and estimate

Threat sources	Threats/ Actions	Repr.	Matched vulnerability
Incompetent user	Unauthorized user gets access to resources	T_1	V_1,V_3, V_7,V_8,V_9 ,V_{10}
	Physical attack	T_2	V_1,V_3, V_7,V_8 ,V_{10}
Hacker	Social engineering	T_3	V_3,V_8 ,V_9
	Tamper the protection relevant mechanisms	T_4	V_1,V_2, V_5,V_6 ,V_{10}
	Reckless network administration	T_5	V_1,V_2,V_3,V_4,V_5,V_9 ,V_{10}
Tactical attack	Improper management	T_6	V_1,V_3, V_4,V_7,V_8
	Viruses, Trojans, Worms	T_7	V_2,V_4, V_6
	Architecture, design and implementation flaws	T_8	V_5,V_6, V_9,V_{10}
	DoS attack	T_9	V_4,V_6, V_7
Industrial Espionage	BoF attack	T_{10}	V_4,V_8, V_9,V_{10}
	No Audits	T_{11}	V_1,V_3, V_4,V_6,V_8 ,V_9,V_{10}
Service administrator	Elevation of privileges	T_{12}	V_1,V_3, V_5,V_7 ,V_8
	Password compromise through plain text communication	T_{13}	V_3,V_4, V_5
	Dictionary/Brute Force attack	T_{14}	V_1,V_3, V_5,V_7 ,V_9
	Arbitrary code execution	T_{15}	V_1,V_4, V_5,V_9

Table 2: Matching threats and vulnerabilities

the likelihood L_{ij} of a threat T_j acting over a vulnerability V_i, as shown in Table 3, i.e., $L_{ij} = \langle V_i, T_j \rangle$ [17, 18, 19].

This likelihood can adopt three values: 0.1, 0.5 and 1, where the value 0.1 represents low likelihood of threat T_j exploiting vulnerability V_i, 0.5 - medium likelihood and 1 - high likelihood. If a threat T_j has no effect on a vulnerability V_i, there is no risk and we make $L_{ij} = 0$.

2.3. Risk Level Analysis

Definition 1: Total Initial Risk (TIR) is the sum of initial risks in an organisation, when no security countermeasure has been applied, and can be computed as follows

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
T1	0.1	0	0.5	0	0	0	0.1	0.1	0	0.1
T2	1	0	0.5	0	0	0	0.1	0.1	0	0.1
T3	0	0	1	0	0	0	0	0.1	0.1	0
T4	0.1	0.1	0	0	0.5	0.1	0	0	0	0.5
T5	0.5	0.5	1	0.5	0.5	0	0	0	0.1	0.1
T6	1	0	0.5	0.1	0	0	0.5	0.1	0	0
T7	0	1	0	0.5	0	0.1	0	0	0	0
T8	0	0	0	0	0.5	0.1	0	0	0.1	1
T9	0	0	0	0.5	0	0.5	0.5	0	0	0
T10	0	0	0	0.1	0	0	0	0.5	0.1	0.5
T11	0.5	0	0.5	0.1	0	0.1	0	0.5	0.5	0.5
T12	1	0	1	0	0.1	0	0.5	1	0	0
T13	0	0	0.5	0.1	0.5	0	0	0	0	0
T14	1	0	0.5	0	0.5	0	1	0	0.5	0
T15	0.1	0	0	0.5	0.5	0	0	0	1	0

Table 3: Likelihood value L_{ij}

$$TIR = \sum_{i=1}^{n} \sum_{j=1}^{m} L_{ij} * I_i, \tag{3}$$

where $TIR \in \mathbb{R}^+$;

Once TIR is known, the organisation becomes aware of how critical identified vulnerabilities are for running a successful business. Thus, the next step is to identify potential security countermeasures that can be applied to reduce the TIR value.

2.4. Control Recommendation

In general, security countermeasures (Table 4) can be categorised as technical, management and operational based on the function they provide. Similar classification can be found in the NIST report [8].

Category	Type	Countermeasure	Representation
Technical	Support	Identification	S_1
		Cryptographic key management	S_2
		Security administration	S_3
		System protection	S_4
	Prevent	Authentication	S_5
		Authorisation	S_6
		Access Control Enforcement	S_7
		Non repudiation	S_8
		Protected communication	S_9
		Transaction privacy	S_{10}
		Audit	S_{11}
	Detect and Recover	Intrusion detection and Containment	S_{12}
		Virus detection and eradication	S_{13}
Management	Preventive	Assign security responsibilities	S_{14}
		Implement separation of duties, least privilege and PC access registration and termination	S_{15}
		Conduct security awareness and technical training and PC access registration and termination	S_{16}
	Detection	Conduct periodic review of security controls	S_{17}
		Periodic system audits	S_{18}
	Recovery	Provide continuity of support and test, maintain it	S_{19}
Operational	Preventive	Control data media access and disposal	S_{20}
		Control software viruses	S_{21}
		Safeguard computing facility (e.g.biometric access control)	S_{22}
		Protect laptops, personal computers, workstations	S_{23}
	Detection	Provide physical security (e.g. motion detectors)	S_{24}

Table 4: A generic list of security countermeasures for identified vulnerabilities and threats

Let each countermeasure be represented as a single bit in the countermeasure vector:

$$\vec{S} = \{S_l\} = \{0, 1\} \, \forall l, \;\; l = 1, 2, ..., k. \tag{4}$$

where S_l represents an individual countermeasure. The value 1 indicates that this countermeasure is applied to the information system and otherwise 0.

The selection of countermeasures is performed by first matching them to identified vulnerabilities as shown in Table 5. In particular, we assign z_{li} based on the characteristics of a certain vulnerability and a countermeasure, and realistically assign the matching values for certain combinations based on its applicability. Previously countermeasure-to-vulnerability matching idea has been proposed in [20].

Information in Table 5 has been mostly retrieved from NIST vulnerability database [14], where general information about vulnerabilities as well as countermeasures from a number of vendors can be found. Furthermore, we have relied on data breach reports [14, 21, 22] and our own knowledge to deliver the concise data about what vulnerabilities can be created or addressed while an appropriate countermeasure is implemented.

Each countermeasure-vulnerability combination z_{li} may have one of the five possible consequences:

$$z_{li} = \begin{cases} 1 & \text{if } S_l \text{ directly addresses } V_i; \\ 0.5 & \text{if } S_l \text{ indirectly addresses } V_i; \\ 0 & \text{if } S_l \text{ and } V_i \text{ do not match;} \\ -0.5 & \text{if } S_l \text{ indirectly creates } V_i; \\ -1 & \text{if } S_l \text{ directly creates } V_i. \end{cases}$$

Note that if it is properly selected, a countermeasure can address a vulnerability, but if it is not adequately chosen may generate a new one. This fact penalises organisations that do not judiciously select countermeasures.

Each of these countermeasures has an associated cost C_l. In this study,

Vulnerability / Countermeasure	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}
S_1	1	0	-0.5	0	0.5	0	-0.5	-0.5	0	-0.5
S_2	0	0	-0.5	0	0.5	-0.5	-0.5	0	0.5	0.5
S_3	1	0.5	-0.5	0.5	1	1	1	1	0.5	0.5
S_4	-0.5	0	-1	0	1	1	1	1	0.5	0.5
S_5	0.5	0	-1	0	0.5	0	0.5	0.5	-0.5	-0.5
S_6	0.5	0	-0.5	0	0	0	0	0.5	0	0
S_7	0	0.5	-0.5	0	0.5	0	0	1	0	0
S_8	0	0	-0.5	0	0.5	0	0	0	0	0
S_9	0	0	0	0	1	0	0.5	0.5	-0.5	0.5
S_{10}	0	0	-1	0	1	0	0.5	0.5	0	-0.5
S_{11}	0.5	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1
S_{12}	0.5	0.5	-0.5	0	0	0	0.5	0.5	1	0.5
S_{13}	0	1	0	0.5	0	0.5	0	0.5	0	0
S_{14}	0	0	0.5	0	0	0	1	0.5	0	0
S_{15}	0	0	0.5	0	0.5	0	0	0	0	0.5
S_{16}	1	0.5	1	0.5	0.5	0.5	1	1	1	1
S_{17}	0.5	0.5	-0.5	0.5	1	0.5	0.5	0.5	1	1
S_{18}	1	1	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5
S_{19}	0.5	0.5	-0.5	0	-0.5	0	0	0.5	0	0
S_{20}	-0.5	0	-0.5	0	0	0	0	0	0	0
S_{21}	0	1	0	0.5	0	0	0	0	0.5	0
S_{22}	1	0	-1	0	0	0	0	0	0	0
S_{23}	0	-0.5	0	0	0.5	0	0.5	0	0	0
S_{24}	-0.5	0	-1	0	0	0	0	0	0	0

Table 5: Matching countermeasures to vulnerabilities z_{li}

we have identified four different costs of implementing a security countermeasure - purchase cost (monetary), operational cost (monetary), training cost (monetary) and man power (monetary).

Purchase cost includes all the costs associated with purchasing a certain countermeasure from a vendor. All the additional sub-charges, if there are some, are also summed up to the total purchase cost value. Operational cost

can be defined as expenses which are related to the operation of a certain countermeasure: this can be fixed or variable costs, such as delivery costs, rent payment or electricity charges. Training cost can be applied for such cases when an additional training is required for an IT staff to increase security awareness. Man power is calculated in persons per hour required to implement a new countermeasure or re-configure the existing one.

	Operational cost ($)	Man power($)	Purchase cost ($)	Training cost ($)	Total ($)
S1	75	48	200	150	473
S2	30	24	0	300	354
S3	150	24	0	200	374
S4	105	24	150	0	279
S5	45	24	50	50	169
S6	30	24	0	0	54
S7	45	48	500	0	593
S8	60	24	0	0	84
S9	30	24	0	0	54
S10	15	24	0	50	89
S11	735	0	200	0	935
S12	15	24	0	0	39
S13	210	24	160	0	394
S14	75	48	0	0	123
S15	60	48	0	0	108
S16	15	24	0	50	89
S17	300	72	100	0	472
S18	150	0	300	0	450
S19	0	48	0	0	48
S20	75	144	500	0	719
S21	420	24	80	0	524
S22	0	72	50	0	122
S23	450	96	160	200	906
S24	15	24	450	0	489

Table 6: Estimated cost value in monetary units

The overall cost for a particular security countermeasure S_l is the sum of the four presented sub-costs defined in monetary units (Eq.5).

$$C_l = \sum_{n=1}^{4} C_n \tag{5}$$

The cost values estimated for each of the countermeasures used in our analysis are shown in Table 6. The values have been estimated relying on the costs offered by the security technology manufacturers and self expertise.

2.5. Risk Assessment and Optimisation Model (RAOM) as an extension of NIST SP800-30

Organisations use risk assessment methodologies to determine the extent of existing threats, vulnerabilities and the risk, associated with the networked systems. NIST SP800-30 risk management guide is designed for organisations which are willing to perform the risk assessment process in a coherent way and thus, help decision makers to quantify the level of risk an organisation has, based on the impact vulnerabilities introduce, likelihood values and overall network state.

Despite the advantages introduced by the NIST SP800-30, there are some limitations to be considered. First of all, it is not designed to help in security countermeasure selection. Also, the data gathering procedure use a simplified way to quantify risk,i.e. by applying a scale to all risk related factors. In overall, the quantitative analysis proposed by the standard cannot be used when optimisation of financial resources in relation to risk is desired. However, the qualitative analysis used to identify all risk related factors can be used as a baseline.

To address the issue covered above, we build on NIST SP800-30, however modify and extend the standard risk assessment procedure by the ad-

ditional methods, denoted as 'Control Selection' and 'Optimisation' (Figure 2). Differently from NIST SP800-30 and other models, we propose a new

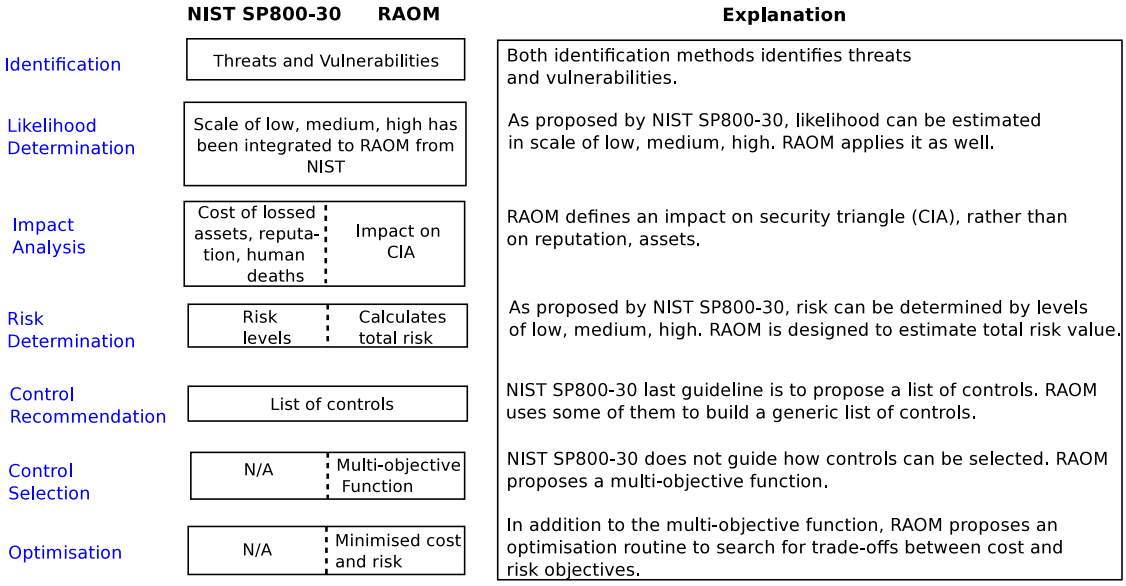


Figure 2: Comparison and extension to the NIST SP800-30

way of quantifying risks in relation to threats and vulnerabilities. From the literature, researchers agree that security is commonly referred to as confidentiality, integrity and availability (CIA) [23]. In fact, a vulnerability will impact CIA, if it is exploited and it will cause disruptions in delivering services to customers, so CIA plays a crucial role in estimating total risks.

Thus, the proposed approach allows to perform a more realistic vulnerability assessment and thus, to calculate the total risk value an organisation holds considering the impact on security triangle.

RAOM also includes a control selection method, which incorporates a multi-objective function and an optimisation technique (Figure 1,Part B). The multi-objective function is proposed considering two conflicting factors:

cost and risk to be optimised. As a result, an optimisation routine can provide with the solutions (trade-offs) that can satisfy organisational security needs in a cost-effective manner.

3. Problem Formulation

We consider two objectives in this study: the total investment cost TC and the risk R. For the $n = 10$ vulnerabilities listed in Table 1 we have suggested $l = 24$ generic security countermeasures (Table 4). As a result, the 2^{24} security countermeasures choices available prove the problem to be hard to solve manually or relying on self-expertise. Furthermore, the time to find the solution increases when the size of the problem increases, i.e., if the number of vulnerabilities n, threats m and countermeasures k increases, the time to find the optimal solution also increases.

Definition 2: Total investment cost

Given a set of k security measures, each having a cost C_l, $1 \leq l \leq k$ and having a vector of $\vec{S} = (S_l)$, $S_l \in \{0, 1\} \forall\, l,\ \ 1 \leq l \leq k$, the total investment cost TC is defined as:

$$TC = \left\{ \sum_{l=1}^{k} C_l S_l : C_l > 0,\ \forall l\ (C_l) \right\} \tag{6}$$

$$S_l = \begin{cases} 1 & \text{if a security measure } l \text{ is selected in the solution;} \\ 0 & \text{otherwise.} \end{cases}$$

Definition 3: Risk

Given a total initial risk TIR, a vector $\vec{S} = (S_l)$, $S_l \in \{0, 1\} \forall\, l,\ \ 1 \leq l \leq k$ and a matching matrix z_{li} , $z_{li} = \langle V_i, S_l \rangle$, the risk R is defined as:

$$R = \left\{ TIR - \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} L_{ij} * I_i * z_{li} * S_l \right\} \tag{7}$$

Problem: *Given a vector of vulnerabilities $\vec{V}$, threats $\vec{T}$ and k security countermeasures, find the vector $\vec{S}$, which minimises total investment cost and risk.*

$$\min_{s_l} [TC,\ R] \tag{8}$$

3.1. Multi-objective Optimisation Principles

In most real world scenarios problems can be formulated to satisfy single or multiple objectives and a decision choice is made based on these objectives and constraints. However, these objectives and constraints are conflicting each other in many cases, making it difficult to find an optimal solution. The conflicting nature of multiple objectives cannot be balanced by just finding a single optimum solution, because when a solution that optimises one of the objectives may not have the same effect on the other objectives. Thus, in case when two or more feasible solutions should be compared, a concept of Pareto front should be considered [24].

Definition 4: Pareto optimal solution, concept of dominance

Let us consider, a minimisation problem, where x and x' are two feasible solutions, X is the set of feasible solutions or decision space, i.e., $x, x' \in X$, p is an objective where $1 \leq p \leq P$, P is the maximum number of objectives, and f_p is the cost function of objective p. Then, solution x strictly dominates or is preferred to solution x' if each cost function value $f_p(x)$ of x is no greater than the corresponding cost function value $f_p(x')$ of x' and at least one cost function value is strictly less: that is, $f_p(x) \leq f_p(x')$ for each p and

$f_p(x) < f_p(x')$ for some p. The set of all non-dominated elements is referred to as non-dominated frontier or a Pareto front [25].

The concept of dominance plays a crucial role for our problem, i.e., minimisation of the security countermeasure cost and risk. A solution that reduces risk will most probably increase cost and vice versa. However, the Pareto front of our problem will provide the optimal trade-offs.

Generating a Pareto set can be computationally expensive, though, a number of stochastic search methods such as evolutionary algorithms, tabu search, simulated annealing have been developed. In general, these methods do not guarantee the optimality of the solution but they often find good approximate solutions. As evolutionary algorithms posses several characteristics that are desired for the multi-objective problems involving multiple conflicting objectives, and intractably large and complex search spaces, these types of search strategies have been successfully used for more than a decade [25].

3.2. Multi-objective Tabu Search (MOTS) for Risk Optimisation

We develop a Tabu Search (TS) technique for solving (8). TS has been applied to a wide range of combinatorial optimisation (e.g. scheduling, routing, traveling salesman) problems. We are now willing to test its the efficiency in the security countermeasure selection problem. The elements, parameters and operation that have been used in our algorithm are presented as following:

- Solution $\vec{S}$.

  A solution is a selection of countermeasures.

- Initial random solution $\vec{S}_{rnd}$.

  The multi-objective TS (MOTS) algorithm starts from creating an

initial solution, which is randomly selected, i.e., each element S_l of solution $\vec{S}$ is set to 0 or 1 with an equal probability.

- The solution space X. This is the set of all possible solutions. The size of X is 2^l, where l is the number of available countermeasures.

- Objective function $f_p(\vec{S})$,
  The objective function $f_p(\vec{S})$ is used to evaluate solution $\vec{S}$ with respect to the objective p. In this case, there are two objective functions, (6) and (7).

- Neighbourhood $\mathcal{N}_s$.
  The TS moves at each iteration from current solution $\vec{S}$ to a neighbouring one $\vec{S}'$ based on a tabu selection process.

- Tabu List (tb).
  The concept of the tabu list is introduced to prevent the problem of possible cycling or/and infinite loop [26]. Tabu list does not allow solutions that have been visited recently.

- Aspiration criteria.
  The aspiration criteria is a global rule for allowing a move, even if it is tabu, if it is a non-dominated solution [27].

- *Stopping criteria.* TS stops iterating when a given condition is reached. The condition could be a given number of iterations, a running time or a solution quality.

When applying MOTS to the minimisation problem proposed in this study, MOTS moves in each iteration from the current solution $\vec{S}$ to a neighbooring one $\vec{S}'$. In our algorithm, neighbooring solutions are always

```
S⃗ = S⃗_rnd; f^c_cur = f_c(S⃗); f^r_cur = f_r(S⃗);        /* random initial solution */
S⃗_best = S⃗;                                          /* initialise best solution */
tb = ∅;                                                /* initialise tabu list */
iter = 0                                               /* set an iteration counter */
while iter ≤ iter_max do
    iter = iter + 1
    neigh = 0                                  /* initialise checked neighboor counter */

    S⃗^best_neigh; f^best_neigh                                  /* best neighboor */

    bestNeighList = ∅                          /* Create best neighboor list */

    while neigh ≤ N_s do
        neigh = neigh + 1
        S⃗′ = randneigh(S⃗)                                  /* neighboor selection */

        f^c_neigh = f_c(S⃗′); f^r_neigh = f_r(S⃗′)            /* evaluate its cost */

        if dominated == 0 then
            /* Is neighboor dominated by some solution in the pareto front?
            */

            S⃗_best = S⃗′; f^c_best = f^c_neigh; f^r_best = f^r_neigh        /* save it */

            StoreSolutionInPareto(S⃗′)
            break;                                 /* stop looking for neighboors */
        end

        if movement(S⃗, S⃗′) in tb then
                                                   /* Is this movement forbidden?  */

            continue;                                          /* Yes, skip it */

        end
        if neigh == 0, f_c(S⃗′) < f_c(S⃗) or f_r(S⃗′) < f_r(S⃗) then
            StoreSolutionInBestNeighList(S⃗′)
        end
        neigh = neigh + 1;
    end
    m = movement(S⃗, S⃗^best_neigh)
    S⃗ = S⃗^best_neigh; f^c_cur = f^{best,c}_neigh; f^r_cur = f^{best,r}_neigh   /* Move to best neighboor */

    tb = tb + [m]                              /* add movement to tabu list */

    removeOld(tabu)                                    /* remove old entries */

end
```

Algorithm 1: Pseudo Code for multi-objective TS

selected randomly by choosing a random countermeasure S_l and changing its allowance from 0 to 1 or vice versa.

First of all, in each iteration the neighboorhood $\mathcal{N}_s$ of a current solution $\vec{S}$ must be defined. In our case, we limit the number of visited neighboors to a value N_s. Thus, MOTS moves from current solution $\vec{S}$ to its best neighbooring one (with the lowest cost and/or risk within the neighboorhood) $\vec{S'} \in N_s$. To construct the Pareto frontier, we remove dominated solutions. The dominated solutions are those, which satisfy the following constraints:

- If the objective function value for cost $f_c(\vec{S})$ is no greater than the corresponding or is equal to cost function value of the neighboor, that is: that is, $f_c(\vec{S}) \leq f_c(\vec{S'})$ and the objective function value for risk $f_r(\vec{S})$ is strictly less than the corresponding risk function value of the neighboor: that is, $f_r(\vec{S}) < f_r(\vec{S'})$;

- If the objective function value for cost $f_c(\vec{S})$ is no greater than the corresponding cost function value of the neighboor, that is: that is, $f_c(\vec{S}) < f_c(\vec{S'})$ and the objective function value for risk $f_r(\vec{S})$ is no greater or equal to the corresponding risk function value of the neighboor: that is, $f_r(\vec{S}) \leq f_r(\vec{S'})$;

- If the objective function value for cost $f_c(\vec{S})$ is equal to the corresponding cost function value of the neighboor, that is: that is, $f_c(\vec{S}) = f_c(\vec{S'})$ and the objective function value for risk $f_r(\vec{S})$ is equal to the corresponding risk function value of the neighboor: that is, $f_r(\vec{S}) = f_r(\vec{S'})$;

It must be noted, that the objective function $f(\vec{S'})$ of the best neighbor does not need to improve the current one $f(\vec{S})$. To avoid getting stuck in a local minima, MOTS may move from current solution $\vec{S}$ to a neighbooring

one $\vec{S}'$ even it is worsening the objective function value [26]. The action of this move from current solution $\vec{S}$ to its best neighboor $\vec{S}'$ is called movement [28]. MOTS stops iterating when a given condition is reached, i.e., given number of iterations. The pseudo code of the MOTS is given in Algorithm 1.

4. Experiments and Discussion

In the following section, we demonstrate the validity of the proposed model by applying an optimisation routine to help decision makers to decide the best solution in multi-objective terms. We compare the qualities of MOTS solutions to optimal ones obtained through the traditional exhaustive search (ES) approah. We examine ten cases when the number of iterations is changed from 500 iterations to 30 000 iterations to examine the speed of the TS approach in finding near optimal solutions.

Prior to presenting actual results, it is appropriate to note that the solving method was written in C++ and executed on an AMD Athlon II X2 245 2.8MHZ processor, 4GB RAM.

4.1. Testing the speed of MOTS

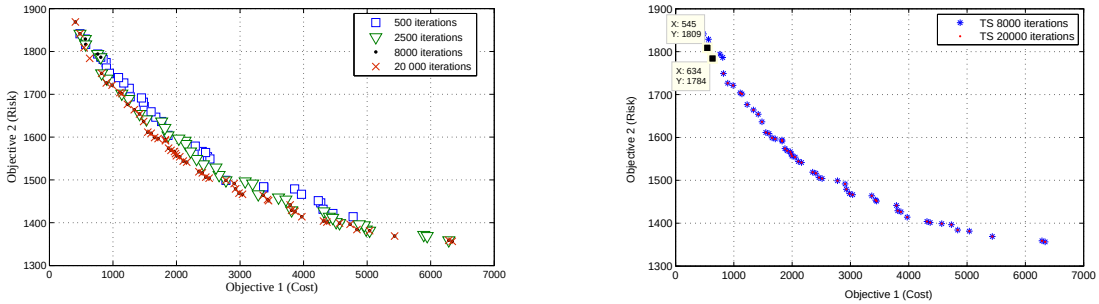
For the first experiment, we test the speed of the MOTS for the original problem. Increasing the number of iterations, we have recorded the time. Table 7 summarizes the efficiency of the MOTS recorded at each case.

The next step of the first experiment was to analyse the quality of solutions obtained. Figure 3(a) shows the non-dominated solutions obtained in 500, 2500, 8000 and 20000 iterations.

In comparison, we took 8000 and 20 000 iteration generated solutions. We did not observe any significant change in the non-dominated solutions

Case	Iterations	Time (s)
1	500	5
2	1000	14
3	2500	44
4	5000	99
5	8000	163
6	10000	221
7	12000	261
8	15000	336
9	20000	450
10	30000	598

Table 7: TS time recorded for ten cases



(a) Convergence of the Pareto Front

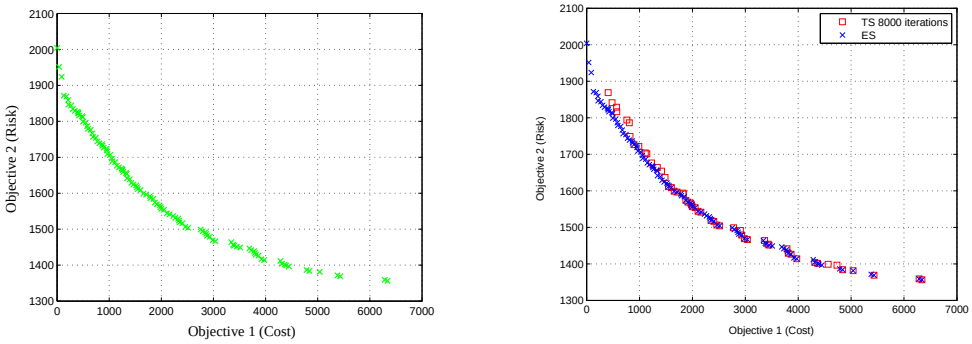
(b) Difference in new solutions

Figure 3: Obtained Pareto front and difference in solutions

by varying the algorithm parameters. In 20000 iterations, MOTS has found the same number of solutions with the difference in four of them. Two of these solutions are labeled with the squared box in Figure 3(b). Once we have noted that variation in solutions is not large and the speed difference is significant for mentioned cases, we can assume, that stopping an algorithm after 8000 iterations the decision maker could get the highest possible number of optimal solutions.

4.2. Testing the quality of solutions

The second experiment was to examine the quality of solutions obtained by MOTS algorithm. We carried it out for the same data set using exhaustive search method (ES). An ES approach was chosen to this problem for several reasons. First, ES is a search technique to solve multiobjective optimisation problems based on enumerative evaluation of each possible solution from a given finite set. Second and perhaps more important, the ES approach is the only way at present to find an exact Pareto Front in multi-objective problems [29].



(a) ES obtained Pareto Front

(b) ES compared to TS

Figure 4: Comparison of the Pareto Front obtained by MOTS and ES algorithms

Figure 4(a) shows the Pareto front obtained by running ES. The algorithm was able to obtain 106 solutions, which surely were optimal ones for this problem. Analysing the quality of solutions, we have compared Pareto fronts obtained by both algorithms, shown in Figure 4(b). We did not see any change in solutions in the intervals of [1000:4500] by the objective 1 (cost) value and [1400:1700] by the objective 2 (risk). A decision maker, in general, would be interested in these intervals, as they are the middle of the

Pareto front with good trade-offs between both objectives.

Despite the fact, that ES is the only algorithm that has an ability to obtain optimal solutions for multi-objective problems, the downside of ES is that the search is computationally expensive.

	TS (8000 iterations)	ES
Time (s)	163	2466
Number of non-dominated solutions	54	106

Table 8: TS and ES comparison data

We have recorded the execution time required to generate the Pareto Front for the ES approach and compared it with the TS 8000 iterations approach (Table 8). MOTS search method has performed 15 times faster than ES. Such a big time difference can be critical in a real life scenario if the decision should be made instantly.

To justify the fact that MOTS has found near optimal solutions, we have calculated Euclidean distance between solutions obtained by both algorithms. Figure 5 shows how close these 54 solutions obtained by MOTS were to the optimum one obtained by ES. It was recorded that 31 solution obtained by MOTS was exactly the same as the ones obtained by ES, thus we can say that MOTS has obtained 30% of optimal solutions when the stopping condition was set to 8000 iterations (Table 9). Other solutions though, are very close to optimum ones, as it can be seen in Figure 5.

4.3. Testing MOTS for the different problem

For the third experiment, we have modified the problem by varying the likelihood L_{ij}, impact I_i, cost C_l and matching z_{li} values. In terms of speed,

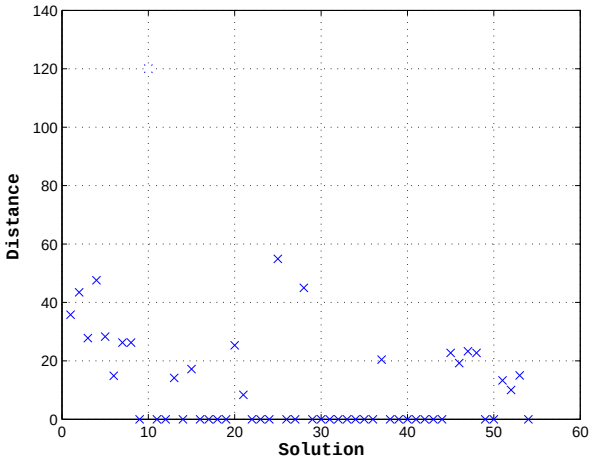


Figure 5: Euclidean distance between 54 MOTS obtained solutions and 106 ES optimum ones

MOTS under different data set has performed very similarly to the original problem. The time and quality of solutions are summarised in Table 9.

From the experiment, we can claim, that in the intervals of 5000 - 10000 iterations the decision maker can obtain higher percentage of optimal solutions ($\sim 30\%$). However, when time is considered as a stopping condition, the best results would be achieved when the algorithm runs between 95s - 200 s.

With such results, MOTS approach shows acceptable levels of accuracy in determining optimal solutions.

Once a decision maker has a better perspective of the solutions possible, the decision on what set of countermeasures should be selected can be justified by the obtained cost and risk trade-offs. The MOTS algorithm has proved to be an efficient way of solving security countermeasure problem when there are two objectives to be minimised.

Iterations	Original Problem (MOTS)			Different Problem (MOTS)		
	Nr.of optimal solutions	Optimality in %	Time (s)	Nr.of optimal solutions	Optimality in %	Time (s)
500	0	0	5	0	0	3
2500	2	1.9	44	17	12	42
5000	11	10	99	25	19	94
8000	31	30	163	34	26	158
10 000	31	30	221	33	25	200
15 000	30	28	336	32	24	317
20 000	29	28	450	31	23	433
ES	106	100	2488	131	100	2842

Table 9: Result comparison under different data sets

5. Conclusion

The importance of decision making in the area of computer security is well understood. Large body of work has been undertaken to support decision makers by providing models which deal with the optimisation of financial investments in relation to computer security. However, most of the models described in existing study are hypothetical rather than practical.

This paper has proposed a novel risk assessment and optimisation model (RAOM), which is partially based on NIST SP800-30 guidelines on performing risk assessments in various organisations. We have adopted the step-by-step procedure of assessing risk, however, we made some important modifications in calculating impact of vulnerabilities and total risk. Due to the fact, that computer security is referred to as CIA, we have designed a way of defining risk in relation to an impact on CIA that each identified vulnerability introduces.

The RAOM differs from previous attempts on improving computer security by applying optimisation techniques in several ways. First of all the

RAOM seeks to assess risk considering an impact on CIA and likelihood that possible threats will exploit identified vulnerabilities, whereas most recent methodologies exclude this realistic fact and risk is assumed to be uniform (e.g.[12]). Moreover, RAOM has an advantage that applying a Tabu Search method to solve a multi-objective countermeasure selection problem formulated in this study makes it possible to review the solutions with the good balance between the two objectives: risk and cost.

Overall it can be concluded that RAOM contributes a new way to make decisions more justified and informed. Experimental results showed that MOTS approach is much faster than the ES approach in searching for the Pareto optimal set. Moreover, the proposed MOTS algorithm showed a good approximation of solutions if compared with the optimal solutions obtained by the ES.

Despite the advantages RAOM and MOTS provides for decision makers, larger size problems (e.g. when the size of security controls, threats and vulnerabilities increases) have not been tested yet. A future research task will thus be to test the performance and scalability of the proposed approach and compare it with other heuristics. Furthermore, we would be interested in adding constraints to the problem, such as a maximum budget assigned for security countermeasure implementation and/or the bounds of risk an organisation is willing to take.

References

[1] T. Neubauer, A. Ekelhart, S. Fenz, Interactive selection of ISO 27001 controls under multiple objectives, in: SEC, 2008, pp. 477–492.

[2] C. Maple, A. Phillips, UK Security Breach Investigations Report, 7Safe (2010).

[3] S. Bistarelli, F. Fioravanti, P. Peretti, Defense trees for economic evaluation of security investments, in: ARES, 2006, pp. 416–423.

[4] H. Lv, Research on network risk assessment based on attack probability, International Workshop on Computer Science and Engineering. 2 (2009) 376–381.

[5] L. Wang, S. Noel, S. Jajodia, Minimum-cost network hardening using attack graphs, Comput. Commun. 29 (2006) 3812–3824.

[6] A. Asosheh, B. Dehmoubed, A. Khani, A new quantitative approach for information security risk assessment, International Conference on Computer Science and Information Technology. (2009) 222–227.

[7] S. Noel, S. Jajodia, B. O'Berry, M. Jacobs, Efficient minimum-cost network hardening via exploit dependency graphs, in: ACSAC, 2003, pp. 86–95.

[8] G. Stoneburner, A. Goguen, A. Feringa, Risk Management Guide for Information Technology Systems, Tech. rep., National Institute of Standards and Technology (2002).

[9] ISO/IEC 27001:2005, Information technology - Security techniques - Information security management systems - Requirements, International Organisation for Standardization (2005).

[10] ISO/IEC 17799:2005, Information technology - Code of practice for information security management (2005).

[11] T. Neubauer, C. Stummer, E. Weippl, Workshop-based multiobjective security safeguard selection, International Conference on Availability, Reliability and Security. (2006) 366–373.

[12] M. Gupta, J. Rees, A. Chaturvedi, J. Chi, Matching information security vulnerabilities to organizational security profiles: a genetic algorithm approach, Decis. Support Syst. 41 (2006) 592–603.

[13] R. Dewri, N. Poolsappasit, I. Ray, D. Whitley, Optimal security hardening using multi-objective optimization on attack tree models of networks, in: ACM Conference on Computer and Communications Security, 2007, pp. 204–213.

[14] NIST, National vulnerability database, automating vulnerability management, security measurement and compliance cheking, `http://nvd.nist.gov/home.cfm`, Accessed before 1st of December 2010.

[15] V. Viduto, C. Maple, W. Huang, An analytical evaluation of network security modelling techniques applied to manage threats, International Conference on Broadband, Wireless Computing, Communication and Applications (2010) 117–123.

[16] C. Maple, V. Viduto, A visualisation technique for the identification of security threats in networked systems, in: Information Visualisation, 2010, pp. 551–556.

[17] Verizon, 2008 Data Breach Investigations Report, Tech. rep., Verizon Business RISK Team (2008).

[18] DTI, Information Security Breaches Survey, Tech. rep., Department of Trade and Industry (2004).

[19] S. Vadera, C. Potter, A.Beard, Information Security Breaches Survey, Tech. rep., PriceWaterHouseCoopers (2008).

[20] S. G. B. H. R. M. J. P. J. R. J. C. R.H. Anderson, P.M. Feldman, Securing the U.S. Defense Information Infrastructure: A Proposed Approach, RAND, Santa Monica, CA, 1999.

[21] M.Templeman, M. Beishon, L. Malachowski, A.Wilson, T. Nash, L. Robertson, Information security - best practice measures for protecting your business, Tech. rep., Department of Trade and Industry (2005).

[22] US-CERT, Introduction to recommended practices, http://www.us-cert.gov/control_systems/practices/, Accessed before 1st of April 2011.

[23] T. Neubauer, C. Hartl, On the singularity of valuating IT security investments, in: ACIS-ICIS, 2009, pp. 549–556.

[24] J. Legriel, C. Le Guernic, S. Cotton, O. Maler, Approximating the pareto front of multi-criteria optimization problems, in: Tools and Algorithms for the Construction and Analysis of Systems, Vol. 6015, Springer Berlin / Heidelberg, 2010, pp. 69–83.

[25] E. Zitzler, M. Laumanns, S. Bleuler, A tutorial on evolutionary multiobjective optimization, in: X. Gandibleux, et al. (Eds.), Metaheuristics for Multiobjective Optimisation, Lecture Notes in Economics and Mathematical Systems, Springer, 2004.

[26] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[27] M. Gendreau, An introduction to tabu search, International series in operations research and management science 57 (2003) 37–54.

[28] D. López-Peréz, Interference avoidance in macrocell-femtocell self-organizing networks: models and optimization, Ph.D. thesis, University of Bedfordshire (2010).

[29] F. Luna, A. J. Nebro, E. Alba, A globus-based distributed enumerative search algorithm for multi-objective optimization, Tech. rep., Departamento de Lenguajes y Ciencias de la Computacion, University of Malaga (2004).