# A Mathematical Derivation of a Risk Assessment Procedure

Marco Benini and Sabrina Sicari*

*Abstract*—**Risk assessment is a well-established engineering practise widely applied on technological systems. Despite its spread, few attempts have been made to formalise its basis in order to provide a non-empirical foundation. In this article we introduce a formal analysis of risk assessment in an algebraic framework, considering also the case of multiple experts. Results about the reliability and the applicability of the framework will be derived according to the structural properties of the problem formalisation.**

***Keywords: Risk analysis, Formal methods, Algebraic modelling***

## 1 Introduction

Security is a process, whose phases have to be implemented in an appropriate order: an important step is the risk assessment task that provides the basis to evaluate the success of the whole process. In fact, it allows to rate on a quantitative basis the security posture of a system, thus the effectiveness of countermeasures.

In this view, risk assessment methods have been developed as "good practises" arising from successful experiences and common sense. There have been few attempts to provide a mathematical formulation to those methods and, as far as we have been able to ascertain, no mathematical foundation has ever been developed. This lack of background is a problem since the outcomes of risk assessment are used in sensitive environments and so there is a strong need for their "certification".

In the case of telecommunication systems, risk assessment is used to protect the network from intrusions and misuses: since most companies have their sensitive information flowing on their internal networks, an abuse of those systems exposes the company to any sort of bad consequences.

In their full generality, risk assessment methods are independent from a specific application domain; nevertheless, few attempts have been made to formalise and to analyse risk assessment in a large, cross-field environment. Since expertise plays an essential role in a successfull application of a risk assessment method, it has been tradition to confine methods within their originating field. Accordingly, although we believe the scope of our results is beyond the originating area, namely security of telecommunication networks, we do not feel confident, that is, we lack the expertise, to claim for a general applicability of our investigations beyond technological systems.

The common approach to risk assessment on technological systems is to find the possible vulnerabilities and to evaluate them according to the damage they may cause and to the probability they will be abused. There are huge databases of vulnerabilities for almost every piece of hardware and software and those repositories are used to test by means of the so-called security scanners, what known vulnerabilities affect a given system. This knowledge is essential for a correct risk assessment, but it is not sufficient. In fact, the correct evaluation of the impact of the found vulnerabilities depends on the structure of the system and on the goals the system should meet. These evaluations are impossible without a direct knowledge of the system and of the environment where the system operates. So, security analysts play an essential role in applying risk assessment methods since their experience is used to evaluate the "local" impact of the possible vulnerabilities.

Because there is no practical way to measure on-the-field the "local" impact of vulnerabilities, most probabilistic approaches suffer from a lack of data, reducing the quality of their outcomes. Hence, although interesting methods based on statistical analyses have been proposed, few of them have been applied in practise. The prominent methods of this sort are briefly discussed in Section 4.

We propose a different way to look at the problem: given one or more experts, how should they proceed in the evaluation of the risk a system is exposed to? In compact terms, we propose to certify the process of risk assessment instead of certifying the results. In this way, the intrinsic dependence on experts' trust is reduced and, to some extent, the outcomes are reproducible. Our work has been previously published in a number of articles [1, 2, 3, 4, 5, 6, 7] where a simple risk assessment method has been proposed, formalised and analysed.

In this paper we want to introduce our risk assessment method as the "minimal" way to perform the risk assessment task when a single expert is asked to analyse a system where vulnerabilities may depend one on another. It will turn out that our method, properly extended, satisfies the requirement of minimality, as shown in Sections 2.1 to 2.3. The extension of the method is

*Manuscript received July 31st, 2009; revised March 16th, 2010.

M. Benini is with the Dipartimento di Informatica e Comunicazione, Università degli Studi dell'Insubria, via Mazzini 5, IT-21100, Varese, Italy (corresponding author to provide phone: +38 0332 218933; fax: +39 0332 218919; e-mail: marco.benini@uninsbria.it).

S. Sicari is with the Dipartimento di Informatica e Comunicazione, Università degli Studi dell'Insubria, via Mazzini 5, IT-21100, Varese, Italy (e-mail: sabrina.sicari@uninsbria.it).

novel and previously unpublished, shading new light to the inner properties of our approach to risk assessment.

More important, in this paper we want to analyse what happens when two or more experts evaluate the same system. Specifically, we want to analyse if it is possible to combine their evaluations in the "most general" way. We have already shown that the combination of evaluations is possible, see [6, 7], but seeking for the most general combination reveals a surprising result: there is no such a thing as the most general way to combine evaluations, intended as the "kernel" of any possible combination. This result is proved in Section 2.4. To show the impossibility to find the most general combination to assess the security posture of a system, given a number of experts' metrics, we use elementary concepts and results of category theory and of the algebra of lattices.

The aims of this article (minimality of the method, impossibility of the most general combination) may appear of a theoretical nature which is not relevant in practise. The toy example of Section 3 shows how it is practically impossible to compare and to combine the evaluations of two security analysts on the small system under consideration. About the practical value of our method, showing that minimality is not a limit, a real case-study has been published in [5].

As a side effect of the main result of this paper, a mathematical framework for risk assessment as a process has been developed. In fact, this is the starting point of our presentation.

## 2 The mathematical framework

A metric is a set of values, not necessarily numbers, used to measure an homogeneous class of observables. Depending on the nature of the observed objects and the goal of the measurement, the values may be combined and compared by means of operations and relations. In the case of risk analysis, the values in a metric form an ordered set, and the relations are equality and less-than-or-equal ($\leq$). Formally, $\leq$ is required to be an order relation, i.e., reflexive ($x \leq x$), transitive ($x \leq y$ and $y \leq z$ implies $x \leq z$) and anti-symmetric ($x \leq y$ and $y \leq x$ implies $x = y$). The $\leq$ relation may be partial, that is, not defined for every pair $x$, $y$ of values. Sometimes, addition of values or scaling (multiplication by a scalar) are used as operations. Usually, when addition is present, it forms an algebraic group. When both addition and scaling are present, the metric forms a module or a vector space, depending on the algebraic structure of scalars (ring or field).

Even in absence of additional ones, the order relation naturally generates a few operations implicitly assumed as given. These operations are the maximum and the minimum of a subset of values or, in the general case of a partial or infinite order, the least upper bound (*lub*, for short) and the greatest lower bound (*glb*).

### 2.1 The formal definition of metric

The operations and the relations on a metric shape its structure: the simplest metrics are formed by a finite set of values with an order relation. Nevertheless, defining a metric just as a partially ordered set prevents the development of risk assessment procedures. In fact, risk is intended to be the worst outcome of an attack to a system, thus the need to calculate the lub of a set of values each one measuring the risk of a single outcome. Also, leveraging among risk evaluations usually requires to compute a glb.

But, to ensure that the lub and the glb exist for every subset of values, the metric must form a complete lattice[1].

**Definition 2.1** *A* lattice *is a partial order $\langle O, \leq \rangle$ such that every pair $x, y \in O$ has a lub, denoted as $x \vee y$, and a glb, denoted as $x \wedge y$. A lattice is* finite *if $O$ is a finite set. Let $U \subseteq O$ be non-empty, then $\bigvee U$ and $\bigwedge U$ are, respectively, the lub and the glb of the elements in $U$, if they exists. A lattice is* complete *if every non-empty subset $U \subseteq O$ has a lub and a glb*[2].

In practise, we are mainly interested in finite lattices. In fact, the set of values used in a risk analysis is always finite: if numbers are used, they have a fixed amount of significant digits: if probabilities are used, only a few decimal places are considered; if qualifiers, like "easy" or "difficult", are used, there is a finite and fixed number of them.

Moreover, we are interested in lattices having two special values, $\bot$ and $\top$, denoting the impossibility to break the system and the immediate ability to abuse of a completely compromised system, respectively. In a metric based on numbers or probabilities, $\bot$ denotes the minimal value and $\top$ the maximal value, while, operating with qualifiers, we assume the existence of two appropriate values[3].

**Definition 2.2** *A lattice $\langle O, \leq \rangle$ is* bounded *if there are two distinct elements $\bot$ and $\top$ such that $\bot = \bigwedge O$, i.e., every element is greater than $\bot$, and $\top = \bigvee O$, i.e., every element is less than $\top$. In a bounded lattice, $\bigvee \emptyset = \bot$ and $\bigwedge \emptyset = \top$.*

**Proposition 2.3** *A finite and bounded lattice is complete.*

*proof:* By induction on the cardinality of subsets: the empty subset has $\bot$ as lub and $\top$ as glb; the glb and the lub of the subset $A \cup \{e\}$ are $e \wedge \bigwedge A$ and $e \vee \bigvee A$, respectively, where $\bigwedge A$ and $\bigvee A$ are defined by induction hypothesis. □

Hence, we a notion of metric that suits our purposes is

**Definition 2.4** *A* metric *is a finite and bounded lattice.*

[1] The theory of lattices is standard. Our treatment is limited to the definitions and the properties of interest in the context of this paper. The interested reader is referred to [8] for a detailed presentation of the mathematical aspects.

[2] Some authors, e.g. [9], do not require the subset $U$ to be non-empty. We follow the general algebraic practise.

[3] This assumption is not committing: it is always possible to add the required values without modifying the inner structure of the lattice.

## 2.2 Modelling an attack

The goal of risk assessment is to determine the likelihood that the identifiable threats of a system will harm, weighting their occurrence with the damage they may cause. A risk assessment method is a procedure to define the risk of the occurrence of one or more threats.

The starting point is to consider a system as a composition of communicating black-box elements; a link between the components $c_1$ and $c_2$, written as $(c_1, c_2)$, means that $c_1$ may directly communicate with $c_2$. Thus, the architecture of the system is modelled by the graph $\mathcal{A} = \langle C, L \rangle$ where $C$ is the set of components and $L$ is the set of links. Moreover, each component or link is assumed to be vulnerable: a vulnerability is a flaw or weakness in the design, implementation or management of a system or component that could be used to violate the security policy, as defined in [10].

The vulnerabilities are organised in a structure showing how they can be used to perform an attack, thought to as a goal to achieve. By recursively dividing each goal into sub-goals, a complex attack can be analysed. The resulting analysis provides a hierarchical plan to perform the attack. This approach is the one of *attack trees* [11, 12], a well-known and widely-adopted method to describe attacks as goals to threaten a system: the attacks are naturally represented in a tree structure, with the main goal as the root node and the different ways of achieving it as children. In turn, each internal node represents an intermediate goal. There are `and` nodes and `or` nodes, each one representing an immediate sub-goal of the father node: `or` nodes are alternative ways to achieve the father goal; `and` nodes represent the steps (ordered from left to right) toward the achievement of the father goal; the leaves of the tree represent the system vulnerabilities.

Thus, the simplest risk assessment method that uses attack trees can be described as follows:

1. The threats to the system under examination are modelled using attack trees and to each vulnerability $v$ is associated a value $\varepsilon(v)$ on a given metric. This value is called the *exploitability* (of the node) and it measures the difficulty to abuse of $v$ and to perform a successful attack.
2. The risk associated to the threat under examination is computed by recursively aggregating the exploitabilities along the attack tree: the exploitability of an `or` sub-tree is the lub the exploitabilities of its children, and the exploitability of an `and` sub-tree is the glb of the exploitabilities of its children.

The aggregated exploitability of the root node measures the feasibility of the attack. Since an attack tree is a finite object, and since the lub and glb operations are associative, it follows that only the binary $\vee$ and $\wedge$ are needed to aggregate the exploitabilities along the attack tree.

Moreover, the finiteness of the attack tree assures that the calculation of the aggregated exploitability terminates with the number of tree nodes as a bound to the number of steps.

The simple method just described assumes no further knowledge on the system than the exploitabilities of the leaves in the attack tree. This method is perfectly adequate when it is possible to evaluate each vulnerability in isolation, as if it does not interfere with the other vulnerabilities. Also, this method implicitly assumes that the experts trying to assess the risk of a system agree both on the possible attack strategies and on the evaluation of each vulnerability.

## 2.3 Modelling dependencies

In most cases, the vulnerabilities of a system are dependent, that is, an attacker can use one of them to simplify the abuse of another one. Thus, there is a relation among the vulnerabilities that specifies how much easier becomes to abuse of the $v$ vulnerability, broken every vulnerability in the set $U$. This relation is called a *dependency* between the ordered pair $(U, v)$ and its weight, denoted as $\varepsilon(v|U)$, measures the exploitability of $v$, given the abuse of $U$[4]. The value $\varepsilon(v|U)$ is called the *conditional exploitability* of $v$ given $U$.

Evidently, when the vulnerabilities are dependent one on the others, the previously defined simple risk assessment procedure is no more sound since the attack tree may not represent all the possible attacks allowing to achieve the root goal from the identified vulnerabilities and following the attack plan.

Given a pair of dependencies $(U, u)$ and $(V, v)$, we say that $(U, u)$ is *stronger* than $(V, v)$ if $u = v$, $U \subseteq V$ and $\varepsilon(u|U) \geq \varepsilon(v|V)$, meaning that it is convenient to abuse of $(U, u)$ than $(V, v)$ since less components have to be violated or the result is easier to obtain. It is worth noticing that $\varepsilon(v) = \varepsilon(v|\emptyset)$. So, $\varepsilon(v)$ must be less than $\varepsilon(v|V)$ for any non-empty $V$ to make the $(V, v)$ dependency significant.

The dependencies can be organised as an hypergraph $\mathcal{D} = \langle W, D \rangle$, where $W$ is the set of all vulnerabilities and $D$ is the set of dependencies. It is safe to assume that, for every $d \in D$, $d$ is not stronger than any other dependency in $D$, since only the strongest dependencies may influence a risk evaluation[5]. Thus, the graph $\mathcal{D}$ is really an hypergraph, having arcs from sets of nodes to a node, but it is not a multigraph. Hence, the cardinality of $D$ is bounded by $|W|2^{|W|}$.

Therefore, the simple risk assessment procedure in Section 2.2 can be extended to consider dependencies.

1. The threats to the system under examination are modelled using an attack tree, as before.
2. The dependencies among identified vulnerabilities are introduced, considering also contextual, architectural and

[4] The chosen notation, $\varepsilon(v|U)$, resembles a conditional probability as $\varepsilon(v)$ resembles a probability. This is done on purpose to help intuition, although the $\varepsilon$ function does not denote a probability measure.

[5] This is true because we perform a worst-case analysis; in an average-case analysis, all the dependencies must be considered.

topological information. The dependencies among vulnerabilities are represented in the *dependency graph* $\mathcal{D} = \langle W, D\rangle$. Moreover, an exploitability value $\varepsilon(u|U)$ weights each dependency $(U, u) \in D$. The values $\varepsilon(u|\emptyset)$ are equal to the initial exploitability of the vulnerabilities, as in the simple method.

3. The exploitability of each vulnerability $v$ is calculated from its dependencies: initially, $\varepsilon_0(v) = \bot$ and then

$$\varepsilon_{i+1}(v) = \varepsilon_i(v) \vee \\ \vee \bigvee \{\varepsilon(v|V) \wedge \bigwedge_{w\in V} \varepsilon_i(w) \colon (V, v) \in D\} \quad (1)$$

whose rationale is to update a value when it is convenient to use the dependency instead of the direct attack pattern. The function $\varepsilon_i$ is said to be *final* when for all $j \geq i$, $\varepsilon_j = \varepsilon_i$.

4. The risk associated to the threat under examination is computed by recursively aggregating the final exploitabilities along the attack tree, as before.

It is not immediately evident that the procedure terminates, i.e., that there exists an index $i$ such that $\varepsilon_i$ is final, although it is clear that the metric must be a complete lattice to apply the method, since the updating rule applies lubs and glbs on subsets of values.

**Theorem 2.5** *There is an index $i$ such that $\varepsilon_i$ is final.*

*proof:* Let $n$ be the number of vulnerabilities in the dependency graph and let $k$ be the cardinality of the metric. Assume that there is no index $i$ such that $\varepsilon_i$ is final. Then, for every $i$, there is a $v$ such that $\varepsilon_{i+1}(v) > \varepsilon_i(v)$. Since $\varepsilon_0(v) = \bot$, after at most $k$ steps, not necessarily consecutive, the exploitability of $v$ becomes $\top$. Also, since at every step a vulnerability is updated, after $n$ steps, not necessarily consecutive, every vulnerability has been updated. After $n \cdot k$ steps every vulnerability reaches the maximum value $\top$ and thus cannot be updated any further. So, at the $n \cdot k + 1$ step, no vulnerability can be updated, contradicting the hypothesis. Thus, there exists an index $i$ such that $\varepsilon_i$ is final and, moreover, $i \leq n \cdot k \leq n^2 2^n$. □

The method just described assumes no further knowledge on the system than the conditional exploitabilities of dependencies, which are fixed. This method is perfectly adequate when it is possible to evaluate each vulnerability in relation to its dependencies and the dependencies do not vary in time. As in the case of the simple risk assessment procedure (Section 2.2), this method implicitly assumes that the experts trying to assess the risk on a system agree both on the possible attack strategies and on the evaluation of dependencies.

## 2.4 Composing metrics

The typical scenario where risk assessment takes place shows a pool of experts each one analysing the system under consideration. These experts work together, exchanging their views and evaluations. Usually, it is possible to obtain a common view of the system architecture and of the possible vulnerabilities. Moreover, usually the experts agree on the possible attacks since this information is construed in a cooperative effort. But, rarely the experts will agree on the evaluations of the impacts of attacks and on the weakness of the various vulnerabilities, since this information directly refers to the competitive nature of their knowledge.

So, it makes sense to model the situation where each expert works on the system with a shared view of the possible attacks and the corresponding vulnerabilities, which amounts to say that each expert agrees to work on the same attack tree and on the same dependency graph[6]. But, due to the differences in experience and knowledge, each expert uses a distinct metric and assigns different exploitabilities to the vulnerabilities.

There is an evident strategy to combine the evaluations of the experts in this case: if their evaluations could be mapped in a *common* metric, then the "worst" evaluation is a measure of the risk associated to a particular threat. In formal terms, this requires to calculate the lub of the values corresponding to the experts' evaluations in the common metric.

There are many possibilities to construct a common metric given a set of metrics. In previous works [6, 7] some of these ways have been introduced and discussed. Here, we want to analyse if it is possible to define a *most general* common metric, that is, a canonical way to combine metrics so that every other way is, in a sense, a specialisation.

Different metrics have common points: the meaning of $\top$ and $\bot$ is always the same; also, the experts may agree on the interpretation of a few values in their metrics ("when I say *hard*, you say 7"). So let us assume to have a correspondence among values that shows what is equivalent in two distinct metrics. The obvious requirement on such a correspondence is soundness, i.e., if $a$ is equivalent to $b$ and $c$ is equivalent to $d$ and $a \leq c$, then it must not be that $d < b$.

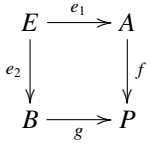
In view of the final outcome we will get, we start from a slightly stronger assumption: the common values form a metric. So, if we have to combine the metrics $A$ and $B$, we assume to have a metric $E$ along with a pair of maps $e_1 \colon E \to A$ and $e_2 \colon E \to B$ whose meaning is that, for every $x \in E$, $e_1(x) \in A$ is equivalent to $e_2(x) \in B$. The use of maps to capture the identified elements suggests that the right instrument to mathematically formalise the problem is Category Theory [13]. In fact, it is immediate to see that the set of metrics and the order-preserving functions respecting $\bot$ and $\top$ is a category.

**Definition 2.6** ***Met*** *is the category whose objects are metrics and whose arrows are the functions $f$ preserving $\leq$, $\bot$ and $\top$, i.e., $x \leq y$ implies $f(x) \leq f(y)$, $f(\bot) = \bot$ and $f(\top) = \top$. So, the arrows preserve the order and the bounds of the metrics.*

In categorical terms, given two metrics $A$ and $B$ where some values are identified via $e_1 \colon E \to A$ and $e_2 \colon E \to B$, our problem

[6] Evidently, it makes sense to consider also the opposite situation. An extension of our results in this direction is under development.

is to find the pushout $P$ (along with the $f$ and $g$ arrows) in the commutative diagram



In fact, when it exists, the pushout of the diagram

$$A \xleftarrow{e_1} E \xrightarrow{e_2} B$$

is, intuitively, the most general object (up to isomorphisms) containing both $A$ and $B$ where the elements in $E$ are identified. The universal nature of the pushout assures that, if it exists, then $P$ is the most general combination of $A$ and $B$, which identifies the values in $E$ in the sense that every other combination $X$ contains $P$ up to a morphism. We will prove that there is no such a most general combination.

There are a few facts about **Met** which turn out to be useful. First, there is a "minimal" metric.

**Proposition 2.7** ***Met*** *has an initial object.*

*proof:* Let 0 be the metric whose domain is $\{\bot, \top\}$ with the only possible order relation. The canonical injection $!\colon 0 \hookrightarrow A$ is the unique function preserving $\bot$ and $\top$; moreover, it trivially preserves the order. □

Second, given two metrics, it is possible to "glue" them in a canonical way that keeps their individual nature.

**Proposition 2.8** ***Met*** *has binary co-products.*

*proof:* Let $A$ and $B$ be metrics, we have to show that there is an object $C$ together with a pair of arrows $j_A\colon A \to C$ and $j_B\colon B \to C$ that is the co-limit of the discrete diagram $A, B$.

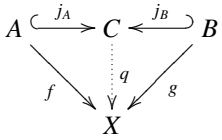
In fact, let $C$ be the disjoint union of $A$ and $B$ where the pairs $(\top_A, \top_B)$ and $(\bot_A, \bot_B)$ are identified and where the order is naturally defined as the union of the given order relations. It is immediate to show that $C$ is, indeed, a metric.

Also, let $j_A$ and $j_B$ be the embeddings of $A$ and $B$ respectively into $C$; these functions trivially preserve $\leq$, $\top$ and $\bot$.

Let $X$ be any metric, and let $f\colon A \to X$ and $g\colon B \to X$. Define $q\colon C \to X$ as

$$q(x) = \begin{cases} f(x) & \text{if } x \in A \\ g(x) & \text{if } x \in B \end{cases}.$$

Because of the definition of $C$, $f(\top) = g(\top) = \top$ and $f(\bot) = g(\bot) = \bot$, so $q$ is well-defined; also, $q$ preserves $\leq$, $\top$ and $\bot$ since $f$ and $g$ do so. Hence $q$ is an arrow in **Met**. Also, being fully determined, it is clear that $q$ is the unique arrow making the following diagram to commute:

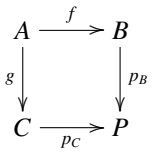


Hence, $C$ along with the canonical injections is the required co-product. It will be denoted as $A \amalg B$. □

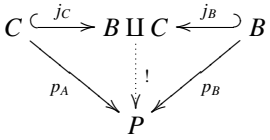
Third, in **Met**, any pushout can be written as the co-equaliser of a co-product. This follows from

**Lemma 2.9** *In a category having initial objects, binary co-products and co-equalisers, every pushout is the co-equaliser of a co-product.*

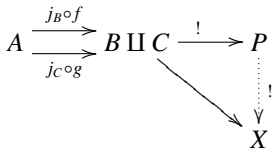
*proof:* Let the following diagram denote a given pushout



Then, being $B \amalg C$ a co-product,



and, applying the universal properties of pushouts,



commutes, thus $P$ and the unique arrow $B \amalg C \to P$ forms the required co-equaliser. □

The last property of **Met** which is required to show the impossibility to construct pushouts in general, is that every pushout in **Met** naturally generates a pushout in **Set**, the category of sets whose arrows are all the functions.

**Lemma 2.10** *Let $F\colon$ **Met** $\to$ **Set** be the forgetful functor, then every pushout $B \to P \leftarrow C$ of the diagram $B \leftarrow A \to C$ in **Met** gives a pushout $F(B) \to F(P) \leftarrow F(C)$ of the diagram $F(B) \leftarrow F(A) \to F(C)$ in **Set**.*

*proof:* The functor $F$ is defined as

- $F(\langle O, \leq \rangle) = O$;
- $F(f\colon \langle A, \leq_A \rangle \to \langle B, \leq_B \rangle) = f\colon A \to B$.

Consider the pushout diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle p_B} \\ C & \xrightarrow[p_C]{} & P \end{array}$$

since $p_C \circ g = p_B \circ f$, it follows by definition of $F$ that $F(p_C) \circ F(g) = F(p_B) \circ F(f)$. So, the same diagram transformed via $F$ commutes in **Set**.

Similarly, the universal property of the diagram in **Met** holds in the transformed diagram in **Set**. Thus, the transformed diagram is a pushout of **Set**. □

Summarising, **Met** has an initial object, binary co-products and every pushout gives raise to a pushout in **Set** via the forgetful functor. By Lemma 2.9, every pushout

$$\begin{array}{ccc} E & \xrightarrow{e_B} & B \\ {\scriptstyle e_A}\downarrow & & \downarrow{\scriptstyle p_B} \\ A & \xrightarrow[p_A]{} & P \end{array}$$

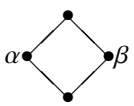
in **Set** is the co-equaliser of a co-product. This amounts to say, see [14], that $P$ is the quotient of the disjoint union of $A$ and $B$ by means of the equivalence relation $e$ identifying the elements of $A$ and $B$ such that $e_A(x) = e_B(x)$, for $x \in E$.

Thus, if the following diagram is a pushout in **Met**

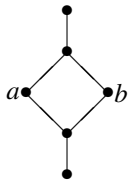
$$\begin{array}{ccc} E & \xrightarrow{e_B} & B \\ {\scriptstyle e_A}\downarrow & & \downarrow{\scriptstyle p_B} \\ A & \xrightarrow[p_A]{} & P \end{array}$$

then $P$ is a complete lattice whose domain is the quotient of the domains of $A$ and $B$ by means of the equivalence relation generated by $e = \{(e_1(x), e_2(x))\colon x \in E\}$. Since every pushout in **Met** is the co-equaliser of a binary co-product, thanks to Lemma 2.9, it suffices to show an example where the co-equaliser does not exist in order to show that **Met** does not have pushouts, in general.

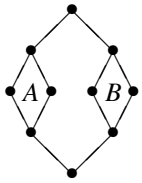
Now, take the following lattice as $E$




and $A$ and $B$ as two copies of the lattice




Thus, $A \amalg B$ is




Let $f\colon E \to A$ be defined as

$$f(x) = \begin{cases} \top & \text{when } x = \top \\ a & \text{when } x = \alpha \\ b & \text{when } x = \beta \\ \bot & \text{when } x = \bot \end{cases}$$

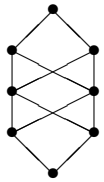
and let $g\colon E \to B$ the same on $B$. Evidently, $f$ and $g$ are arrows in **Met**.

Now, the diagram

$$E \underset{j_B \circ g}{\overset{j_A \circ f}{\rightrightarrows}} A \amalg B$$

does not have a co-equaliser in **Met**. In fact, such an object $Q$ must be a co-equaliser also in **Set**, thus it must be that $Q = (A \sqcup B)/r$, the quotient of the disjoint union of the domains of $A$ and $B$ by means of the equivalence relation $r$, defined as $r = (e \cup e^{-1})^*$, where $e = \{(f(x), g(x))\colon x \in E\}$. Now, the only possible order relation over $Q$ which allows a function $q\colon A \amalg B \to Q$ preserving the order of $A \amalg B$ is given by the relation $\leq_Q = \leq_{A \sqcup B} /r$ as easily proved in elementary algebra, since necessarily $q(x) = [x]_r$, the equivalence class with respect to $r$ containing $x$.

But $\langle Q, \leq_Q \rangle$ is



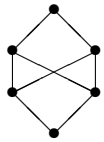
so it is not a metric, not being a lattice.

Hence, **Met** has no pushouts, or, in the original problem, it is not always possible to find the most general combination of two metrics, given a sound identification of values.

Since we have shown that $P = (A \sqcup B)/r$ with $r = (e \cup e^{-1})^*$ and $e = \{(e_A(x), e_B(x))\colon x \in E\}$ is the pushout of $A$ and $B$ via $E$ in **Set**, and since $\langle Q, \leq_Q \rangle$ with $\leq_Q = \leq_{A \sqcup B} /r$ is a finite and bounded order, our negative result says that $Q$ is not always a lattice.

A natural question is whether it is possible to impose conditions on $A$, $B$ and $E$, or, alternatively, on $e_A\colon E \to A$ and $e_B\colon E \to B$, such that $\langle Q, \leq_Q\rangle$ is a lattice, thus forming the required pushout in **Met**.

**Proposition 2.11** *Let $T$ be the finite and bounded order whose Hasse diagram is*



*Then, there is $t\colon T \to P$ mono if and only if $P$ is not a lattice.*

*proof:* Suppose $P$ is a lattice. If there is such a map $t$, $P$ contains $T$ as a suborder, which is clearly impossible since the strict lower sets of the images of the topmost values just below the top are equal while the values differ.

Vice versa, being $P$ a finite and ordered set, there are $q_1, q_2 \in Q$ distinct with at least two different lubs. So, we can find a suborder of $Q$ having as points $\bot, q_1, q_2, \top$ and the two lubs, which is the image of $T$ via an appropriate definition of $t$. □

Considering the counterimage of $T$ in $A \amalg B$ via $q\colon x \mapsto [x]_r$, it is easy to see that a co-equaliser in **Set** must equate a pair of uncomparable values in $A$ with a pair of uncomparable values in $B$. Thus a "fork" must exist in $E$. Hence, a sufficient condition to force $P$ to be a lattice is to require $E$ to be a linear order.

Moreover, applying the proposition in a different way, there must be a pair of forks in $P$ sharing the same base and whose apexes are distinct, so that in $A \amalg B$ the counterimages of the base points are equated by $q$ while the counterimages of the apexes are not. Thus, a sufficient condition to ensure $P$ to be a lattice is to require that every lub and glb of uncomparable values in $A$ and $B$ must be in the image of $e_A$ and $e_B$, respectively. Formally, for every $x, y \in A$ such that $x$ and $y$ are uncomparable in $A$, $(x \wedge y) \in e_A(E)$ and $(x \vee y) \in e_A(E)$, and analogously for $B$.

## 3 An Illustrating Example

In this section, we show a simple example that illustrates the major points we developed so far. Despite its simplicity, it resembles a real situation, allowing for a deeper insight on the meaning of our findings.

Consider a train company with an automatic system to sell tickets. A traveller can buy a ticket from the ticket dispenser, an automatic vending machine, located inside the train station. The dispenser is just a dedicated station to access the company web site. The dispenser is able to sell a ticket, given the destination, the departing time and the name of the traveller plus his/her credit card. Moreover, the dispenser allows to modify a sold ticket given the traveller name and the ticket number, eventually asking for an additional fee or refunding the difference. A traveller is allowed to modify any of the traveller's name, the destination and the departing time.

Before a traveller can access a train, s/he has to pass a ticket check. There, an human employee asks the traveller for his/her ticket and checks whether it is valid; also, the traveller must exhibit a document to prove his/her identity.

The company is satisfied by this system since it proves to be very economical and secure, but it worries about people trying to take the train without paying the ticket. This may happen, as the company has experienced, because some last-minute jumpers wait until the train is about to depart and then, with some trick, they try to convince the employee that they are late so to have a light check of their invalid ticket. To solve this problem, the company asks Alice and Bob, two security analysts, to assess the risk associated to the event "someone gets the train but no one has paid his/her ticket".

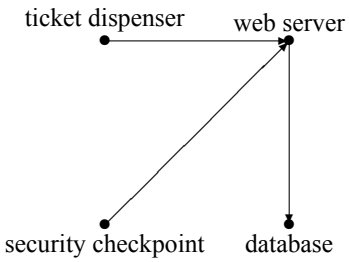



Figure 1: The system architecture of the example.

**Goal**: get the train without paying
- 1: Cheat the employee at the security checkpoint
  - 1.1: Give a false identity ($V_1$)
  - OR 1.2: Corrupt the employee ($V_2$)
- OR 2: Break the web server security at the dispenser
  - 2.1: Inject code ($V_3$)
  - OR 2.2:Guess another traveller's ticket number and name ($V_4$)

Figure 2: The attack plan for "Take the train without paying for the ticket."

After some work, Alice and Bob are able to draw the system architecture, in Figure 1, and a reasonable attack plan, in the form of an attack tree, depicted in Figure 2. The reason behind the attack plan is simple: the company claimed "a ticket is valid if the name of the traveller is correct, the destination is correct and it has been paid, we do not care who pays!". So, Alice and Bob checked that it is impossible to get a new ticket without paying, hence a new ticket from the dispenser is always valid. The only way to misuse a new ticket is to steal it and to give a false identity at the security checkpoint. A traveller without a ticket can take a train only if s/he passes the security check by corrupting the employee. Mr. George Neverpay may try to modify a valid ticket by guessing in some way the traveller's name and the ticket number and then giving his name and his destination to the dispenser,

so to get a reissue of the ticket. Alternatively, George may try to cheat the web server beside the dispenser, trying to create a new ticket in the database. To this aim, he can access the dispenser by modifying a very economical ticket and trying to put something strange (code injection) in the form on the web page.

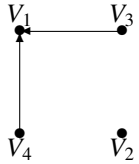



Figure 3: The dependencies in the example.

Since forcing the security of the web server from the ticket dispenser simplifies to show a new identity to the security gate, the dependency graph is as in Figure 3.

Now, Alice and Bob have to evaluate the vulnerabilities and the dependencies: they agree that the human trust is incomparable with the web server security, but they disagree on the way to measure these aspects of the system.

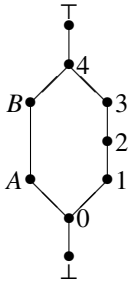



Figure 4: Alice's metric

Alice uses a security scanner to trace the vulnerabilities of the technological parts of the system: the security scanner provides a value for each vulnerability in the integer range 0 to 5. Alice decides that 0 means "secure up to current knowledge", so she uses 5 as the $\top$ of her metric, while 0 is a value just above $\bot$. She decides that human beings are not as trustable as computers (a common attitude among security analysts), so she decides to use the value $A$ for a trustable human, the value $B$ for a non-trustable human, and 4 for a combination of non-trustable human using a non-trustable computer. The resulting metric is shown in Figure 4.

Using her metric, Alice states that her initial exploitabilities are $\epsilon_0(V_1) = A$, $\epsilon_0(V_2) = 0$, $\epsilon_0(V_3) = 0$ and $\epsilon_0(V_4) = 2$. She evaluates that the conditional exploitabilities are $\epsilon(V_1|V_4) = 4$ and $\epsilon(V_1|V_3) = 4$. Applying up to the fixpoint the formula (1), she finds that $\epsilon(V_1) = 4$, $\epsilon(V_2) = 0$, $\epsilon(V_3) = 0$ and $\epsilon(V_4) = 2$, and thus the aggregated evaluation of the attack goal is 4. The reader is invited to notice how dependencies influenced the final evaluation.

Bob uses three levels of human risk ($H_1$, $H_2$ and $CH_3$) and three levels of computer risk ($C_1$, $C_2$ and $CH_3$); the highest value $CH_3$ is common and it happens when a risky individual uses a non-trustable computer. Since things can always go worse, he decides

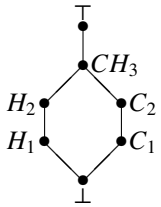



Figure 5: Bob's metric

that $\top$ is above the common value; also, he is convinced that a level of risk is never null, so $\bot$ is below both minima in his personal metric. The result is depicted in Figure 5.

Bob states that the exploitabilities of the system vulnerabilities are $\epsilon_0(V_1) = H_2$, $\epsilon_0(V_2) = H_1$, $\epsilon_0(V_3) = C_1$ and $\epsilon_0(V_4) = C_2$. He evaluates that the conditional exploitabilities are $\epsilon(V_1|V_4) = H_2$ and $\epsilon(V_1|V_3) = H_2$, in fact, choosing to apply the simple risk assessment procedure, thus the initial evaluations are also the final ones. Hence, the aggregated exploitability of the root goal is $CH_3$. The reader is invited to notice that the way to obtain the simple risk assessment procedure from the one using dependencies is general.

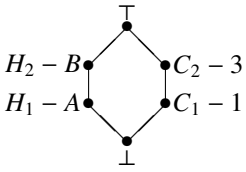
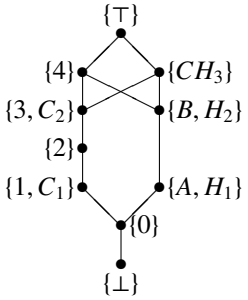



Figure 6: How Alice and Bob compare their metrics.

Since Alice and Bob agree on the evaluation principle that human and computers are incomparable, they identify the "pure" values in their metrics. Specifically, the value $X - Y$ in $E$, the identification metric shown in Figure 6, means that the value $X$ in Bob's metrics is identified with the value $Y$ in Alice's. Evidently, $\top$ and $\bot$ are always identified.

A few calculations say the co-equaliser in **Set** is $\{\{\bot\}, \{0\}, \{1, C_1\}, \{2\}, \{3, C_2\}, \{4\}, \{A, H_1\}, \{B, H_2\}, \{CH_3\}, \{\top\}\}$, and the corresponding partial order is




which is not a lattice since, e.g., the lub of $\{3, C_2\}$ and $\{B, H_2\}$ is not unique.

Combining the previous evaluations, one calculates the lub of the exploitabilities obtained by Alice and Bob for each vulnerability.

The result is $\epsilon(V_1) = \{4\}$, $\epsilon(V_2) = \{A, H_1\}$, $\epsilon(V_3) = \{1, C_1\}$ and $\epsilon(V_4) = \{3, C_2\}$. But, if the initial evaluation of Alice on $V_2$ would have been 1, her final $\epsilon(V_2)$ would have been 1, as well; in this case, the combination would not have been possible, since, in the combined structure, the lub of 1 and $H_1$ is either $\{4\}$ or $\{CH_3\}$, due to the fact that the combination of Alice and Bob's metrics is not a metric. Hence, the limiting result on the combination of metric has a practical value since it applies also to our experts' evaluations.

## 4 Related works

In literature there are many attempts to face the risk assessment problem; some of them define systematic approaches while others provide more ad-hoc methods to evaluate the likelihood of (a class of) violations. Even though the application of risk evaluation methodologies has been widely discussed and analysed, see, e.g., [15, 16, 17, 18], among information security experts there appears to be no agreement regarding the best or the most appropriate method to assess the possibility of computer incidents [19].

In particular, we have found of interest Baskerville's [20] description of the evolution of various ad-hoc methods to measure risk that sometimes could be combined to improve the accuracy of the security evaluation.

On the side of systematic approaches, S. Evans et al. [21] present a system security engineering method to discover system vulnerabilities and to determine what countermeasures are best suited to deal with them: the paradigm of this work is *analysing information systems through an adversary's eyes*.

Differently, [22] provides a probabilistic model that measures security risks. It is possible to calculate risk starting from hybrid values of a quantitative and/or qualitative nature.

With respect to the previous works, our approach, starting from its initial definition in [1], has been based on the structured evaluation of single vulnerabilities along with their mutual dependencies. In this respect, the results in [21] are similar to ours, although they do not propose a formal method based on mathematical arguments. In fact, the distinctive aspect of our work with respect to the discussed ones is the mathematical formalisation of the risk assessment method in order to derive its characterising properties. Also, the use of hybrid values in [22] resembles our approach to metrics as algebraic structures, even though, we do not map them down to probabilistic estimates.

There are more formalised approaches in literature, employing a graph-based representation of systems and their vulnerabilities, that provide methods whose properties are, at least partially, mathematically analysed. Among those approaches, of prominent interest are those based on attack graphs [23, 24], where state-transition diagrams are used to model complex attack patterns. In particular, [23] proposes the use of attack graphs to automate the step of hardening a network against a multi-step intrusions. The proposed security solution is expressed as an adjustable network configuration rather than a set of countermeasures to possible exploits.

Similarly, [25] divides a system into sub-domains and each subdomain could be characterised by vulnerabilities. Applying probability theory and graph transformations [25] evaluates the possibility that a insecurity flow exploits some vulnerability to penetrate into the system. The extreme consequence of this family of approaches is to use model-checking techniques to simulate attacks, like in [24].

In this respect, our approach is simpler both in the method and in its formalisation. Despite its simplicity, our results are stronger on the mathematical side and some experimentation [2, 4, 5] make evident the practical value of the method in real-world situations. In fact, we use the attack tree model [11, 12] to evaluate the security threats combining them with the dependency graph, a formalisation of a piece of experts' knowledge. This combination is part of the subject of our mathematical analysis, and being a richer structure than the simple attack trees, we are able to derive stronger properties for our method [7].

On a rather different comparison line, the software component paradigm in software engineering has received a great deal of interest from both industry and academia since it allows the reusability of components and a natural approach to distributed programming. A software component is independently developed and delivered as an autonomous unit that can be combined to become part of a lager application.

Despite its evident benefits, the component interdependence is often ignored or overlooked [26], leading to incorrect or imprecise models. In order to avoid this problem, complete models should be specified taking into account system interconnections. In agreement with this point of view [26, 27, 21, 22, 19] present models for assessing security risks taking into account interdependence between components.

Particularly, [26] uses techniques for automating and enhancing risk assessment studies of technological processes using qualitative models. A set of fundamental parameters and primitive functions are defined for the domain from which the system behaviour is derived, detecting a number of interesting interdependencies among components.

Similarly, [27] defines a model based on security policy and individual risks. The model gives the possibility to evaluate if the risk associated to each transaction is acceptable. The evaluation of risk also takes into account context information.

As a matter of fact, independently from their application areas, the risk assessment methods have a core weakness: the use of subjective metrics. In fact, in the scientific community the main criticism to these methods is about the fact that values are assigned on the basis of personal knowledge and experience. In extreme cases, these assessment are regarded as *random* values, making the total risk evaluation process to be considered as a *guess*.

It is a fact that the evaluation metric behind exploitability deeply influences the risk evaluation. But, at least in our treatment, what matters is the *structure* of the metric rather than its absolute value. Generalising, in many field of ICT there is the need to define an objective metric. In the abstract, a metric is defined as [28] the instrument to compare and to measure a quantity or a quality of an observable.

Our treatment of metrics follows the work of N. Fenton, in particular [29]. In agreement with him, we consider measurement as the process by which values are assigned to attributes of entities, in our case to the exploitability of a vulnerability. Therefore, even though there is no widely recognised way to assess risks and to evaluate the induced damages, there are various approaches that provide methodologies by which the risk evaluation becomes more systematic.

In particular, Sharp et al. [19] developed a scheme for probabilistic evaluation of the impact of the security threats and proposed a risk management system with the goal of assessing the expected damages due to attacks in terms of their economical costs. Z. Dwaikat et al. [27] defined security requirements for transactions and provided mechanisms to measure likelihood of violation of these requirements.

Looking toward risk assessment as a decision support tool, Fenton [30] proposed the use of Bayesian networks. He distinguishes between certain and uncertain criteria and points out the power of Bayesian networks to reason about uncertainty. Differently, our approach toward objective risk assessment is based on the abstraction over values, thus what matters in our treatment is the *structure* of the metrics. Hence, objectivity is gained by considering values in the metric not as *absolute measures of risk*, but, instead, as *relative evaluations*. Therefore, in agreement with [26, 21, 30, 22], the information computed by our model can be used as a decision support.

## 5 Conclusions

This paper wanted to introduce a mathematical framework to justify risk assessment methodologies. In this sense, the methods we presented are *minimal*, that is, the simplest one can conceive given our assumptions. The assumptions we started from are generally accepted by the engineering community as shown in the related works.

It is relevant to notice that our results have been compared with works outside the original application area of our method. In particular, our claim as stated in the Introduction, that our investigations can be generally applied to technolgical systems is supported by the comparisons with the related works in, e.g., software engineering.

Although some improvements have been made on our previous findings, see [3, 4, 5, 6, 7], the novel result and the main contribution of this paper is the negative answer to the question "there exists the most general combination of two metrics given a sound identification of values between them?" To derive this result a more mathematical approach has been adopted, showing that the category of metrics does not have all the co-limits and interpreting the desired combination as a push-out, a specific instance of co-limits. This result required a neater reformulation and a better understanding of the formal definition of our risk assessment method. This deeper and refined analysis has been presented here for the first time. Also, the use lattice algebra and category theory to handle evaluation metrics is innovative in the field of risk assessment methodologies, as far as the authors' know.

In our opinion, the development of risk assessment methods inside strong mathematical frameworks is the most promising way to study them as more than mere "good practises". In this sense, our results show a first concrete step in this direction.

A possible future development is to study the application of our mathematical framework on different technological fields, e.g., industrial engineering, or on human-oriented areas, like medicine. We hope that some researcher with an expertise in those fields may be interested in extending our findings.

## Acknowledgements

We wish to thank Prof. R. Melen of Università degli Studi di Milano-Bicocca for the fruitful discussions about some of the preliminary results.

## References

[1] S. Sicari, D. Balzarotti, and M. Monga, "Assessing the risk of using vulnerable components," in *Quality of Protection. Security Measurements and Metrics*, D. Gollmann, F. Massacci, and A. Yautsiukhin, Eds. New York, NY, USA: Springer-Verlag, Jun. 2006, pp. 65–78.

[2] M. Benini and S. Sicari, "Risk assessment: Intercepting VoIP calls," in *Proceedings of the VIPSI 2007 Venice Conf.* Belgrade, SB: IPSI Belgrade, Mar. 2007.

[3] M. Benini and S. Sicari, "A mathematical framework for risk assessment," in *New Technologies, Mobility and Security*, H. Labiod and M. Badra, Eds., The 1st International Conf. on New Technologies, Mobility and Security. Netherlands: Springer, Mar. 2007.

[4] M. Benini and S. Sicari, "Assessing the risk to intercept VoIP calls," *Journal of Computer Networks*, vol. 52, no. 12, pp. 2432–2446, 2008.

[5] M. Benini and S. Sicari, "Risk assessment in practice: A real case study," *Computer Communications*, vol. 31, no. 15, pp. 3691–3699, 2008.

[6] M. Benini and S. Sicari, "Towards more secure systems: How to combine expert evaluations," in *Proceedings of the 4th International Conf. on Security and Privacy in Communication Networks*. New York, NY, USA: ACM, 2008, pp. 1–6.

[7] M. Benini and S. Sicari, "Risk assessment via partial orders," *Advances in Computer Science and Engineering*, vol. 3, no. 1, pp. 19–46, 2009.

[8] G. Grätzer, *General Lattice Theory*, 2nd ed. Basel, CH: Birkhäuser, 2003.

[9] P. Johnstone, *Stone spaces*, ser. Cambridge studies in advanced mathematics. Cambridge, UK: Cambridge University Press, 1982, vol. 3.

[10] R. Shirey, "RFC 2828: Internet security glossary," May 2000. [Online]. Available: `http://www.ietf.org/rfc/rfc2828.txt`

[11] A. Moore and R. Ellison, "Survivability through intrusion-aware design," CERT Coordination Center, Tech. Rep. 2001-TN-001, 2001.

[12] B. Schneier, "Attack trees," *Dr. Dobb's Journal*, vol. 24, no. 12, pp. 21–29, 1999.

[13] S. Mac Lane, *Categories for the Working Mathematician*, 2nd ed., ser. Graduate Texts in Mathematics. Heidelberg, DE: Springer, 1997.

[14] R. Goldblatt, *Topoi: The Categorial Analysis of Logic*. Mineola, NY, USA: Dover, 2006.

[15] F. den Braber, T. Dimitrakos, B. Gran, M. Lund, K. Stølen, and J. Aagedal, "The CORAS methodology: Model-based risk management using UML and UP," in *UML and the Unified Process*, L. Favre, Ed. Hershey, PA, USA: IRM Press, 2003, pp. 332–357.

[16] C. Alberts, A. Dorofee, J. Stevens, and C. Woody, "Introduction to the Octave approach," Oct. 2003. [Online]. Available: `http://www.cert.org/octave/approach_intro.pdf`

[17] B. Jenkins, "Risk analysis helps establish a good security posture; risk management keeps it that way," pp. 1–16, 1998, white paper. [Online]. Available: `http://www.nr.no/~abie/RiskAnalysis.htm`

[18] T. Siu, "Risk-eye for the IT security guy," pp. 1–20, Feb. 2004. [Online]. Available: `http://www.giac.org/certified_professionals/practicals/gsec/3752.php`

[19] G. Sharp, P. Enslow, S. Navathe, and F. Farahmand, "Managing vulnerabilities of information system to security incidents," in *Proceedings of the 5th International Conf. on Electronic Commerce*. New York, NY, USA: ACM Press, 2003, pp. 348–354.

[20] R. Baskerville, "Information system security design methods: Implications for information systems development," *ACM Computing Surveys*, vol. 25, no. 4, pp. 375–412, 1993.

[21] S. Evans, D. Heinbuch, E. Kyle, J. Piorkowski, and J. Wallener, "Risk-based system security engineering: Stopping attacks with intention," *IEEE Security & Privacy Magazine*, vol. 2, no. 6, pp. 59–62, 2004.

[22] M. Sahinoglu, "Security meter: A practical decision-tree model to quantify risk," *IEEE Security & Privacy*, vol. 3, no. 3, pp. 18–24, May/June 2005.

[23] S. Noel, S. Jajoidia, B. O'Berry, and M. Jacobs, "Efficient minimum-cost network hardening via exploit dependency graphs," in *Proceedings of 19th Annual Computer Security Applications Conf.* Los Alamitos, CA, USA: IEEE Computer Society, 2003, pp. 86–95.

[24] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing, "Automated generation and analysis of attack graphs," in *Proceedings of the 2002 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 273–284.

[25] I. Moskowitz and M. Kang, "An insecurity flow model," in *Proceedings of the 1997 Workshop on New Security Paradigms*. New York, NY, USA: ACM Press, 1997, pp. 61–74.

[26] G. Biswas, K. Debelak, and K. Kawamura, "Application of qualitative modelling to knowledge-based risk assessment studies," in *Proceedings of the 2nd International Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, M. Ali, Ed., vol. 1. New York, NY, USA: ACM Press, 1989, pp. 92–101.

[27] Z. Dwaikat and F. Parisi-Presicce, "Risky trust: Risk-based analysis of software system," in *Proceedings of the 2005 Workshop on Software Engineering for Secure Systems — Building Trustworthy Applications*. New York, NY, USA: ACM Press, 2005, pp. 1–7.

[28] S. Arshad, M. Shoaib, and A. Shah, "Web metrics: The way of improvement of quality of non web-based systems," in *Proceedings of the International Conf. on Software Engineering Research and Practice*, H. R. Arabnia and H. Reza, Eds., vol. 2. Las Vegas, NV, USA: CSREA Press, 2006, pp. 489–495.

[29] N. Fenton, "Software measurement: A necessary scientific basis," *IEEE Transactions on Software Engineering*, vol. 20, no. 3, pp. 199–206, 1994.

[30] N. Fenton and M. Neil, "Making decisions: Bayesian nets and MCDA," *Knowledge-Based Systems*, vol. 14, no. 7, pp. 307–325, Nov. 2001.