A Mathematical Framework for Risk Assessment

Marco Benini and Sabrina Sicari

Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria
via Mazzini 5, IT-21100, Varese, Italy
{marco.benini, sabrina.sicari}@uninsubria.it

Abstract. Risk assessment is an important step in the development of a secure system: its goal is to identify the possible threats to a system, their impact and, henceforth, to evaluate the connected risks. Although several systematic approaches have been developed to perform a risk assessment task, the current methodologies rely on the quantitative evaluations of experts in a substantial way. This paper addresses the problem of detaching the methodology results from the subjective judgements of experts, by formalising a risk assessment methodology in an appropriate mathematical framework that reduces the subjective aspects in experts' evaluations.

1 Introduction

Despite the fact that risk assessment is a well-established engineering practice to evaluate the security status of a complex system, the significance of the obtained results is often debated [1] since it depends on the quantitative judgements of one or more human experts and, thus, the results are said to be influenced by the subjective views of the experts.

The present work wants to cope with this problem by developing a mathematically formalised risk assessment procedure whose dependence on the judgements of human experts is greatly reduced and controlled.

The core of our proposal lies in considering the quantitative measures given by the experts as *relative*: the measures are considered to be part of a *metric*, i.e., an organised system of possible values, and the meaning of a single value lies in its relationship with the *structure* of the whole metric.

In this way, a sensible risk assessment methodology, like the one we are going to discuss, relies only on the structure of the metrics. The immediate effect is that the evaluations of different experts using each one a different metric, can be compared and integrated as far as the metrics are compatible.

Henceforth, the need for a mathematical formalisation becomes clear: a mathematical framework allows to define the notion of metric, its structure and the idea of compatible metrics, and, furthermore, it allows to prove that a risk assessment does not depend on the choice of a metric, but, instead, essentially the same result is obtained by using any compatible metric.

Therefore, in Section 2, we introduce the risk assessment methodology and, in Section 3, we describe the mathematical framework allowing to prove its correctness, i.e., its ability to obtain a result. In Section 4, we show an example that clarifies our claim that the risk assessment procedure is independent from the values in the metric and that compatible metrics will produce equivalent results. Finally, in Section 5, a comparison with the existing approaches is shown, allowing the reader to position our result in the existing research streams.

2 The risk assessment methodology

In general, 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 possibly cause. Therefore, a risk assessment methodology is a procedure whose outcome is an estimation of the risk connected with the occurrence of one or more threats.

In this work, we analyse the risk assessment methodology introduced in [2]; specifically, in this Section, we illustrate the methodology, briefly discussing its foundations, while in Section 3, we will introduce its main properties and we will use them to derive some interesting facts about the quality of the risk evaluations the methodology produces.

In general, the risk is measured by a function r of two variables: the damage potential of the hazard and its level of exploitability. The damage potential is often defined as the average loss of money an attack may cause, although, in our approach, any other kind of sensible measure can be used. Moreover, the level of exploitability is a measure of the difficulty to make an attack including both the easiness and the reproducibility of an attack, as defined, e.g., in the STRIDE/DREAD theory [3].

The methodology evaluates the total risk of a threat by means of a sequence of steps described as follows:

1. The threat to the system under examination is modelled by using an attack tree [4, 5]: the attack goal is the root node and its children nodes represent possible ways of achieving it. Recursively, the children can be alternative subgoal, each one satisfying the father goal (`or` subtrees) or partial subgoals, whose composition satisfies the father goal (`and` subtrees). The leaves of the tree are the *potential* vulnerabilities that should be matched with the *actual* vulnerabilities of the system. To each vulnerability v is associated an index $E_0(v)$, called its *exploitability*, which measures the difficulty to exploit v in order to perform a successful attack.
2. The dependencies among the identified vulnerabilities are introduced: a vulnerability A depends on a vulnerability B if, when B is already exploited, then A becomes easier to exploit. Moreover, each dependency is weighted by an exploitability value, using the same metric as E_0.
3. The final exploitability of each single vulnerability is calculated taking into account its initial value, which does not consider dependencies, and its dependencies, according to the algorithm described in Section 2.1.

4. The risk associated to the threat under examination is finally computed by recursively aggregating exploitabilities along the attack tree. The exploitability of an `or` subtree is the easiest exploitability of its children, and the exploitability of an `and` subtree is the most difficult exploitability of its children. The aggregated exploitability measures the level of feasibility of the attack and is combined with the damage potential to finally assess the risk of the threat.

The *metric* employed in the evaluation of exploitabilities and their dependencies is the set of possible values for $E_0(v)$. We require this set to be a partial order: this choice reflects the difficulty to compare an arbitrary pair of vulnerabilities in order to decide their relative difficulty; usually, similar vulnerabilities are easily compared, while different vulnerabilities may be compared only to some extent, e.g., saying that both are easier or more difficult to exploit than a third one.

Evidently, it is safe to suppose that the metric contains a finite number of elements, since the vulnerabilities of the system are always finite, and, similarly, it is safe to assume that the partial order contains a maximum, denoted as 1, and a minimum, denoted as 0. This is justified since every actual vulnerability is easier to exploit than to violate the ideal perfectly secure component, while each vulnerability is harder to exploit than the ideal perfectly insecure component.

In practice, a meaningful assessment of $E_0(v)$ is a matter of both experience and ingenuity, but, as explained in Section 3, just the *relative* exploitability has to be estimated.

2.1 Exploitability of dependent vulnerabilities

The system is formalised as a graph $\mathcal{A} = \langle C, L \rangle$ where C is the set of *components* and L is the set of *links* between components. The components and the links are exposed to the set of vulnerabilities V_C and V_L, respectively, where an element $(u, v) \in V_C$ means that the component u is susceptible to be subverted thanks to the flaw v. Therefore, the set of the system's vulnerabilities is $V = V_C \cup V_L$.

Initially, during the step 1 in our methodology, an expert assesses how easy and repeatable is to exploit every single vulnerability to gain the control of a component or a link in the given architecture. We call this the initial exploitability $E_0(v)$ of the vulnerability v in the system $\mathcal{A}$.

In general, the functions $E_i : V \mapsto \mathcal{O}$ map vulnerabilities to $\mathcal{O}$, a partial ordered set of degrees of exploitability. The functions are indexed by a step number (details later), thus the E_0 function generates the exploitability values $E_0(v)$. The $\mathcal{O}$ set, modelling the expert's metric, is a finite, partially ordered set containing two distinct elements, 0, its minimum, and 1, its maximum, as already explained in the preceding Section.

However, the architecture of the system imposes dependencies among vulnerabilities. For example, we need to understand if it is easier to exploit a vulnerability of a component given that an input link attached to it has already been compromised or a component attached to any of its input links has already been violated. We denote with $E(v|w)$ the exploitability of v given that the vulnerability w has already been exploited.

The dependencies among vulnerabilities are represented in the *dependency graph* $\mathcal{D} = \langle V, D \rangle$, whose nodes are the vulnerabilities and the edge (w, v) is in D iff $E(v|w) \geq E_0(v)$, i.e., an edge (w, v) means that it is easier to compromise an element suffering the v vulnerability when one has already compromised an element affected by the w vulnerability.

The number of exploitability evaluations is bounded since the number of edges in the $\mathcal{D}$ graph is, at most, $|V|(|V| - 1)$. However, in practice, most of the vulnerabilities are usually independent, and the evaluations the expert has to *guess* is typically closer to $|V|$ than to $|V|^2$.

Initially, the value associated to each node v in the dependency graph $\mathcal{D}$ is $E_0(v)$, that is, the initial measure of how difficult is to exploit the vulnerability. The conditional exploitabilities are used to label the edges they belong to. The conditional values are derived in step 2 of our methodology.

The initial assessment depicted in the graph $\mathcal{D}$ does not take into account that each vulnerability could be exploited thanks to the previous exploitation of one of the vulnerabilities on which it depends. Therefore, the labels of the nodes are iteratively updated by considering the easiest way, i.e., the maximum value, to exploit an incoming vulnerability in the dependency graph. In turn, each incoming vulnerability could be exploited by controlling the affected element or leveraging on the dependency itself: the most difficult, i.e., the minimum, constraints the value. Therefore, the update rule for labels is defined by the following formula:

$$E_{i+1}(v) = \sup(\{E_i(v)\} \cup \{\inf\{E(v|w), E_i(w)\} \colon (w, v) \in D\}) \ . \qquad (1)$$

Thus, the third step of the methodology consists in iteratively applying (1) for each vulnerability, until the system converges to an equilibrium after a suitable number n of steps. Then, the values of $E_n(v)$ represent the final exploitability of each vulnerability v considering also its dependencies.

3 The mathematical framework

As the reader may expect, the choice of the metric $\mathcal{O}$ influences the results obtained in the application of the methodology. On a more subtle level, the mathematical characters of the methodology depend on the *structure* of the ordering $\mathcal{O}$, as we are going to clarify in this Section.

The first property of the method is its *convergence*; we want to prove that the methodology guarantees to reach an equilibrium in the computation of the exploitability values. A side effect of the proof is that the equilibrium is a reached in a number of steps bounded by $|\mathcal{O}||V|$, being V the set of vulnerabilities.

Theorem 1. *Given a dependency graph $\mathcal{D} = \langle V, D \rangle$, there is a number k such that, for every $v \in V$, $E_{k+1}(v) = E_k(v)$.*

Proof. We notice that, for any number i and for any $v \in V$, $E_{i+1}(v) \geq E_i(v)$ because of the definition of exploitability. Moreover, we know that the exploitability values form a finite partial order $\mathcal{O}$: let n be the number of elements in $\mathcal{O}$.

Therefore, before reaching the equilibrium, for every step i, there is a vulnerability v such that $E_{i+1}(v) > E_i(v)$. This situation can occur only n times per vulnerability, since, after at most n updates, the value $E_i(v)$ reaches the maximum of the metric, thus it cannot be updated anymore. Hence, after at most $n|V|$ steps, every vulnerability must be stable. □

Corollary 1. *Given a dependency graph $\mathcal{D} = \langle V, D\rangle$, there is a number k such that, for every $v \in V$ and $i, j \geq k$, $E_i(v) = E_j(v)$.*

It is important to remark that Theorem 1 provides an upper bound to the number of iterations: as obvious by the use of the pigeon hole principle, this bound is unnecessarily large, and in practice, convergence is usually obtained in a few steps.

In our observation, it resulted that the methodology enjoys another interesting property, that reveals how the iterative calculation depends only on the structure of the ordering of metrics. Specifically,

Theorem 2. *Given a dependency graph $\mathcal{D} = \langle V, D\rangle$ and two metrics $\mathcal{O}_a = \langle O_a, \leq_a, 0_a, 1_a\rangle$ and $\mathcal{O}_b = \langle O_b, \leq_b, 0_b, 1_b\rangle$, if $g \colon \mathcal{O}_a \to \mathcal{O}_b$ is a morphism[1] from $\mathcal{O}_a$ to $\mathcal{O}_b$ such that $g(E^a_0(v)) = E^b_0(v)$ for every $v \in V$ and $g(E^a(v|w)) = E^b(v|w)$ for every $(w, v) \in D$, then, for any $v \in V$ and for any i, $g(E^a_i(v)) = E^b_i(v)$, where E^a and E^b are the exploitability functions using, respectively, $\mathcal{O}_a$ and $\mathcal{O}_b$ as metrics.*

Proof. By induction on i, the number of steps; the base step is obvious by hypothesis, while the induction step is as follows:

$$\begin{aligned}
&g(E^a_{i+1}(v)) = \\
&g(\sup_a(\{E^a_i(v)\} \cup \{\inf_a\{E^a(v|w), E^a_i(w) \colon (w, v) \in D\})) = \\
&\sup_b(\{g(E^a_i(v))\} \cup \{\inf_b\{g(E^a(v|w)), g(E^a_i(w)) \colon (w, v) \in D\}) = \\
&\sup_b(\{E^b_i(v)\} \cup \{\inf_b(E^b(v|w), E^b_i(w) \colon (w, v) \in D\})) = \\
&E^b_{i+1}(v) \enspace .
\end{aligned}$$

□

Some comments are due:

- The hypothesis "$g(E^a_0(v)) = E^b_0(v)$ for every $v \in V$ and $g(E^a(v|w)) = E^b(v|w)$ for every $(w, v) \in D$" encodes the fact that the initial situation the method is applied to, is the same, modulo the g morphism.
- The g is a morphism, i.e., a function respecting the relation and the constants of the order. The meaning of this requirement is that the metrics are compatible, that is, any comparable pair of values in the first metric is associated with a pair of values in the second metric that gets compared in the same way.

[1] A morphism is a structure-preserving function. In particular, a function f between $\mathcal{O}_a$ and $\mathcal{O}_b$ is a morphism iff for every $x, y \in O_a$ such that $x \leq_a y$, it holds that $f(x) \leq_b f(y)$.

3.1 An illustrating example

This section tries to clarify our findings by means of an abstract example that evidences the core of our results without the complexities of a real case study.

The scenario is as follows: we have two security experts, Alice and Bob, working together to evaluate the risk of a network attack to a complex system. They developed a suitable attack tree (not shown) and they agree both on the set of vulnerabilities affecting the system, and on the way they depend one on each other. Hence, our experts produce the system dependency graph, whose nodes are the identified vulnerabilities and whose arcs are the dependencies.

In practice, the depicted scenario is common: the possible ways to conduct an attack, the identification of the vulnerabilities and, finally, the dependencies among the identified vulnerabilities are subjects on which experts can easily integrate their knowledges, thus producing a common, agreed picture of the security status of a system.

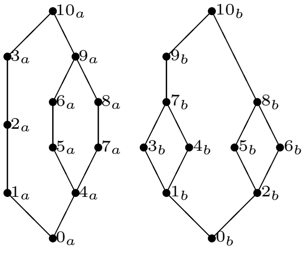

Fig. 1. The metrics a and b

Differently, when the experts are asked to quantify the risks connected to the identified vulnerabilities, their evaluations may diverge because of the application of different metrics. In our example, Alice adopts the metric a while Bob uses the metric b; both of them are represented in Fig. 1. The drawing shows the minima (0_a and 0_b) at the bottom, the maxima (10_a and 10_b) at the top, and a value x is less than y if x is below y and connected to. The supremum of two elements x and y is the minimal point above x and y, connected to both of them, and, dually, the infimum of x and y is the closest connected point below them.

In the scenario, Alice develops an initial evaluation of the exploitability values, synthesised in Fig. 2; Bob does the same, as illustrated in Fig. 3. These evaluations are the result of the application of the experts' experience and judgement, thus, at least to some extent, the values are subjective.

Applying our methodology, Alice and Bob can calculate the final risk assessment, considering also the role of dependencies: after a few iterations of the

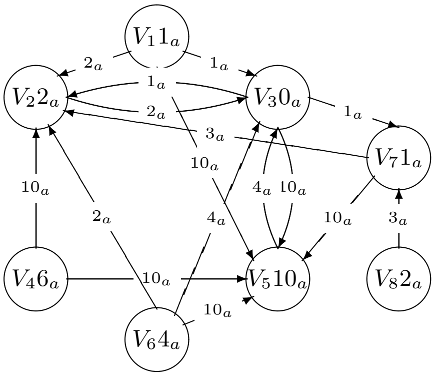

Fig. 2. The initial evaluation of Alice

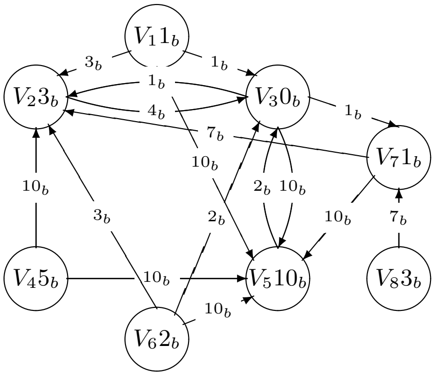

Fig. 3. The initial evaluation of Bob

application of (1), Alice derives the following risk vector

$E^a(V_1)$	$E^a(V_2)$	$E^a(V_3)$	$E^a(V_4)$
1_a	10_a	10_a	6_a

$E^a(V_5)$	$E^a(V_6)$	$E^a(V_7)$	$E^a(V_8)$
10_a	4_a	2_a	2_a

,

while Bob obtains as his final result

$E^b(V_1)$	$E^b(V_2)$	$E^b(V_3)$	$E^b(V_4)$
1_b	10_b	10_b	5_b

$E^b(V_5)$	$E^b(V_6)$	$E^b(V_7)$	$E^b(V_8)$
10_b	2_b	3_b	3_b

.

It is evident that the derived evaluations are different, so we expect the measured risk to differ. Nevertheless, following the Theorem 2, if the metrics are compatible, then the results coincide, modulo a *renaming* of the values in the metrics.

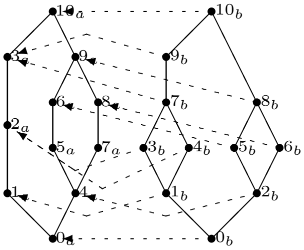

Fig. 4. The morphism from the metric b to a

The *renaming* function is the morphism relating the metrics of our experts, and its existence is the criterion to say that the metrics are compatible. For example, the metric b used by Bob can be mapped in the metric a of Alice via the morphism g shown in Fig. 4: it is immediate to check that, if $x < y$ in the metric b, then $g(x) < g(y)$ in the metric a.

If one transforms the resulting risk vector of Bob by means of the g morphism, the result is

$g(E^b(V_1))$	$g(E^b(V_2))$	$g(E^b(V_3))$	$g(E^b(V_4))$
1_a	10_a	10_a	6_a

$g(E^b(V_5))$	$g(E^b(V_6))$	$g(E^b(V_7))$	$g(E^b(V_8))$
10_a	4_a	2_a	2_a

,

that is exactly the result of Alice.

Henceforth, Theorem 2 says that, the values in a metric have a conventional meaning which is determined up to morphisms between compatible metrics.

4 Related works

Even though the application of risk management methodologies has been widely discussed and analysed, see, e.g., [6–9], among information security experts there appears to be no agreement on the best or the most appropriate method to assess the probability of computer incidents [10].

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.

In particular, we have found of interest Baskerville's description [11] 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, O. Sami Saydjari et al. [12] present a system security engineering methodology to discover system vulnerabilities, and to determine what countermeasures are best suited to deal with them: the paradigm that synthesises this work is *analysing information systems through an adversary's eyes.*

With respect to the previous works, our approach, starting from its initial definition in [2], has been based on the structured evaluation of single vulnerabilities along with their mutual dependencies. In this respect, the results in [12, 13] are similar to ours, although they do not propose a formal methodology 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.

In this respect, there are more formalised approaches, employing a graph-based representation of systems and their vulnerabilities, that provide methodologies whose properties are, at least partially, mathematically analysed. Among those approaches, of prominent interest are those based on attack graphs [14, 15], where state-transition diagrams are used to model complex attack patterns. The extreme consequence of this family of approaches is to use model-checking techniques to simulate attacks, like in [15].

In comparison, our approach is simpler both in the methodology and in its formalisation. Despite its simplicity, our results are stronger on the mathematical side and some experimentation [16] make evident the practical value of the method in real-world situations.

In fact, we use the attack tree model [4, 5] to evaluate the security threats combining them with the dependency graph, a formalisation of an essential piece of knowledge of experts. This combination is the subject of our mathematical analysis, and dealing with a richer structure than the simple attack trees, we are able to derive stronger properties.

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

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 values, thus limiting the previous criticism.

Generalising, in many field of ICT there is the need to define an objective metric. In the abstract, a metric is defined [17] as 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 [18].

In agreement with him, we consider measurement as the process by which numbers or symbols 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.

Looking towards risk assessment as a decision support tool, Fenton [19] proposed the use of Bayesian networks. Differently, our approach towards objective risk assessment is based on the abstraction over values, thus what matters 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 of risks*. Therefore, in agreement with [20, 12, 19, 21], the information computed by our model can be used as a decision support.

5 Conclusions

This work addressed the problem of formalising a risk assessment procedure. The problem behind the formalisation is that a risk assessment depends on the quantitative evaluation of exploitabilities, a task performed by human experts, and, thus, subjective.

The idea developed here to cope with this problem, is to abstract over the metrics, i.e., the values used to quantify the threats, in order to force the risk assessment methodology to depend on the structure of a metric, instead of depending on its absolute values. In this way, two experts adopting different but compatible metrics will produce similar evaluations, that is, evaluations that are different in their form, but equivalent in their meaning.

In this respect, we formalised in an appropriate mathematical framework a simple, yet effective risk assessment methodology, and we have proven inside the framework its correctness, i.e., its ability to produce the expected evaluations, in Theorem 1. Moreover, we claimed that the methodology depends only on the structure of the employed metrics, and we gave a precise mathematical formalisation of this statement. As a side effect, we have shown a formal characterisation of the notion of compatible metrics in terms morphisms between partial orders.

References

1. Redmill, F.: Risk analysis: A subjective process. Engineering Management Journal **12**(2) (April 2002) 91–96
2. Sicari, S., Balzarotti, D., Monga, M.: Assessing the risk of using vulnerable components. In Gollmann, D., Massacci, F., Yautsiukhin, A., eds.: Quality of Protection. Security Measurements and Metrics, New York, NY, USA, Springer-Verlag (June 2006) 65–78
3. Howard, M., Leblanc, D.: Writing Secure Code. Microsoft Press (2003)
4. Moore, A., Ellison, R.: Survivability through intrusion-aware design. Technical Report 2001-TN-001, CERT Coordination Center (2001)
5. Schneier, B.: Modelling security threats. Dr. Dobb's Journal (December 1999)
6. Alberts, C., Dorofee, A., Stevens, J., Woody, C.: Introduction to the Octave approach (October 2003)
7. den Braber, F., Dimitrakos, T., Gran, B., Lund, M., Stølen, K., Aagedal, J.: The CORAS methodology: Model-based risk management using UML and UP. In Favre, L., ed.: UML and the Unified Process. IRM Press (2003) 332–357
8. Jenkins, B.: Risk analysis helps establish a good security posture; risk management keeps it that way (1998) White paper.
9. Siu, T.: Risk-eye for the IT security guy (February 2004)

10. Sharp, G., Enslow, P., Navathe, S., Farahmand, F.: Managing vulnerabilities of information system to security incidents. In: ICEC '03: Proceedings of the 5th International Conference on Electronic Commerce, New York, NY, USA, ACM Press (2003) 348–354
11. Baskerville, R.: Information system security design methods: Implications for information systems development. ACM Computing Survey **25**(4) (1993) 375–412
12. Evans, S., Heinbuch, D., E.Kyle, Piorkowski, J., J.Wallener: Risk-based system security engineering: Stopping attacks with intention. IEEE Security & Privacy Magazine **2**(6) (2004) 59–62
13. Moskowitz, I., Kang, M.: An insecurity flow model. In: NSPW '97: Proceedings of the 1997 Workshop on New Security Paradigms, New York, NY, USA, ACM Press (1997) 61–74
14. Noel, S., Jajoidia, S., O'Berry, B., Jacobs, M.: Efficient minimum-cost network hardening via exploit dependency graphs. In: ACSAC '03: Proceedings of 19th Annual Computer Security Applications Conference, IEEE Computer Society (2003) 86–95
15. Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.: Automated generation and analysis of attack graphs. In: SP '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy, Washington, DC, USA, IEEE Computer Society (2002) 273–284
16. Benini, M., Sicari, S.: Risk assessment: Intercepting VoIP calls. In: Proceedings of the VIPSI 2007 Venice Conference. (March 2007) To appear.
17. Arshad, S., Shoaib, M., Shah, A.: Web metrics: The way of improvement of quality of non web-based systems. In Arabnia, H.R., Reza, H., eds.: SERP '06: Proceedings of the International Conference on Software Engineering Research and Practice. Volume 2., CSREA Press (2006) 489–495
18. Fenton, N.: Software measurement: A necessary scientific basis. IEEE Transactions on Software Engineering **20**(3) (1994) 199–206
19. Fenton, N., Neil, M.: Making decisions: Bayesian nets and mcda. Knowledge-Based Systems **14**(7) (November 2001) 307–325
20. Biswas, G., Debelak, K., Kawamura, K.: Application of qualitative modelling to knowledge-based risk assessment studies. In Ali, M., ed.: IEA/AIE '89: Proceedings of the Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. Volume 1., New York, NY, USA, ACM Press (1989) 92–101
21. Sahinoglu, M.: Security meter: A practical decision-tree model to quantify risk. IEEE Security & Privacy **3**(3) (May/June 2005) 18–24